نظم إدارة قواعد البيانات

الجزء الثاني

DBASE III+ DBASE IV FOXBASE+ FOXPRO

م . مصطفى رضا عبد الوهاب
د . علاء الدين محمد فهمى
مصطفى محمد إسماعيل
أ .د. محمد فهمى طلبه
أ.د. عمرو جنيد
أ.د. محمد على الشرقاوى

٦

موسوعة دلتا كمبيوتر

نظم إدارة قواعد البيانات

رقم الإيداع ١٩٩١/٢٦٥٣

مطابع المكتب المصري الحديث
MODERN EGYPTIAN PRESS

تقـــــديم

ان التطور في لغات البرمجة للحاسب الالكتروني قد مر بمراحل متعددة أرتبطت بعدة عوامل من أهمها التطور التكنولوجي المتلاحق في مكونات الحاسب المادية وما تبع ذلك من زيادة سرعة عمليات الحاسب المختلفة وزيـادة كفاءتها . وقد أدى هذا الى تطور هائل في نظم التشغيل وظهور مفاهيم جديدة مثل تعدد الوظائف وتعدد المستخدمين واستخدام الشبكات بالاضافة الى العديد من الخصائص التى تميز الاجيال المتلاحقة من الحاسب الالكترونى .

ومن الملامح الرئيسية لتطور لغات البرمجة البعد التدريجي عن التدخل فى العديد من العمليات التفصيلية التى تتم بواسطة الحاسب لأداء مهمة معينة مما أتاح الفرصة أمام مخططى البرامج لاستخدام خصائص الحاسب دون الحاجة الى الالمام الكامل بعملياته الداخلية . ومن المعروف أن الحاسب حقيقة لا يفهم الا لغة الواحد والصفر وهى ما يطلق عليها لغة الماكينة (Machine Language) . وكان مخطط البرامج لا يستطيع التفاهم مع الحاسب الا من خلال هذه اللغة المعقدة . ثم ظهرت لغات أكثر سهولة قامت بتكوين مجموعات من الواحد والصفر فى رموز بسيطة واستخدام هذه الرموز فى كتابة البرامج . وتسمى هذه اللغات باللغات الرمزية (Symbolic Languages) أو لغات التجميع (Assembly Languages) . وأخذت اللغات فى التطور مع زيادة درجة التمثيل حتى وصلت الآن الى لغات الجيل الرابع (4th Generation Languages) حيث أتاحت هذه اللغات لمخطط البرامج التعامل مع النوافذ والقوائم الواضحة التى يستطيع من خلالها تصميم البرامج المطلوبة بسهولة تامة ويسر .

وقد انعكس هذا التطور الكبير على المتخصصين فى مجال الحاسب حيث أصبح على مخططى البرامج متابعة كل جديد فى مجال نظم تطوير البرامج وأدواتها المتقدمة وذلك حتى يمكنهم الاستفادة من خصائصها في تصميم النظم المتميزة التى توفر الكفاءة العالية وسهولة الاستخدام . ومن ناحية أخرى فقد أصبح على المستخدم ضرورة الالمام بهذه البرامج التطبيقية الحديثة حتى يستطيع الاعتماد على نفسه فى الاستفادة منها والاضافة اليها . وقد أدى ذلك الى ظهور جيل جديد من المستخدمين الذين يمتلكون خبرة كبيرة فى التعامل مع العديد من البرامج التطبيقية الى جانب القدرة على تصميم النظم الخاصة بهم .

وقد كان لهذا التطور فى مجال الحاسبات أثره فى طبيعة الكتب المتخصصة التى ظهرت على المستوى العالمي فى الآونة الاخيرة. فبعد أن كانت هذه الكتب - الى وقت قليل مضى - تركز على الجوانب النظرية ، أصبحت الآن تركز على أساليب استخدام التطبيقات وعلى تقديم الخبرات والمهارات العملية لمستخدمى الحاسبات .

وقيام مؤسسة "دلتا" بتقديم موسوعتها الجديدة فى تكنولوجيا وعلوم الحاسب يعوض النقص الشديد الذى تعانى منه المكتبة العربية فى هذا المجال حيث أن معظم الكتب العربية الموجودة ليست سوى ترجمة أو تلخيص سطحى لدليل التشغيل للنظم المختلفة بينما يحتاج المستخدم الى توضيح الكثير من الجوانب العلمية والفنية

بالاضافة الى خصائص تشغيل النظم . وهذا الجهد من مؤسسة دلتا هو امتداد لسياستها الواعية واحساسها بمسئوليتها نحو التطور التكنولوجى بالمنطقة العربية .

وهذا الكتاب هو أحد كتب موسوعة "دلتا" لتكنولوجيا و علوم الحاسب والذى يمثل حلقة الاتصال بين الجوانب التطبيقية والجوانب العلمية والفنية . فبالرغم من تركيزه على شرح خصائص تشغيل نظم عائلة (DBase) ، الا أنه اهتم بتوضيح مفهوم قواعد البيانات بصفة عامة وتوضيح العمليات المرتبطة بها مثل الفهرسة والبحث عن البيانات وأساسيات تصميم البرامج ، و ..... الخ .

ومما لاشك فيه أن موضوع هذا الكتاب يعد من أهم الموضوعات التى تشغل أذهان جميع المتخصصين فى مجال الحاسب وذلك لارتباطه المباشر بالمجالات العملية سواء للأفراد أو للمؤسسات . فقد أصبحت قواعد البيانات وتطبيقاتها تغطى معظم مجالات الحياة وأصبح استخدام الحاسب فى ادارة البيانات والسيطرة عليها أمرا مألوفا فى كل موقع . ورغم توفر عدد كبير من البرامج التطبيقية التى تخصصت فى ادارة قواعد البيانات ، مثل (Informex) ، (Oracle) ، (Paradox) ، (Focus) ، .... الخ الا أن مؤسسة "دلتا" اختارت برامج عائلة (DBase) لتكون موضوع هذا الكتاب بأجزائه الثلاث وذلك لما تتمتع به عائلة (DBase) من الشيوع والانتشار وسهولة الاستخدام بالاضافة الى قدرتها على التعامل مع العديد من نظم التشغيل .

ا. د. محمد فهمى طلبه

محتويات الكتاب

مقدمــــة

هذا الكتاب هو الجزء المكمل للكتاب الاول الذى نشر تحت اسم " نظم ادارة قواعد البيانات " - الجزء الأول . ورغم أن الكتاب الأول يتضمن الشرح التفصيلي لخصائص برامج عائلة (DBase) بالاضافة الى أساسيات كتابة البرامج بواسطتها وهذا الشرح يعتبر كافيا للالمام الكامل بالخصائص الفنية لهذه البرامج الا أن مؤسسة "دلتا" رأت استكمالا لهذا الجزء أن توفر للمستخدم مرجعا شاملا لجميع الأوامر والدوال المستخدمة فى عائلة (DBase) . متضمـــنا الأوامر والدوال الخـــاصة ببرامج (DBase III +) ، (DBase IV +) ، (FoxBase +) ، (FoxPro) .

وهذا المرجع يعتبر ضروريا لمن يريد اكتساب خبرة كبيرة ببرامج عائلة (DBase) والاعتماد على نفسه فى تصميم النظم الكاملة باستخدام برامج العائلة . حيث أن المستخدم أثناء اعداده للبرامج يحتاج دائما الى الرجوع الى أمر معين ومعرفة الشكل (Syntax) الخاص به بالاضافة الى خصائص استخدام هذا الأمر . وهذا المرجع الشامل يتيح له الوصول مباشرة الى الأمر المطلوب . حيث أن الأوامر مرتبة حسب الترتيب الهجائى للحروف . وسوف يجد القارىء الشرح الوافى لكل أمر متضمنا الرسم التوضيحى الملائم والأمثلة المناسبة .

ونظرا لأن هذا الكتاب هو جزء مكمل للكتاب الأول كما سبق الايضاح ، فقد تم ترتيب الأبواب بتسلسل مكمل للتسلسل الخاص بالكتاب السابق . لذلك يبدأ الكتاب بالباب رقم (٢٧) . مع ملاحظة أن الأوامر والدوال مرتبة حسب الترتيب الهجائى للحروف الانجليزية .

الباب السابع والعشرون

أهم الأوامر المستخدمة

ان قـائمة الأوامر المستخدمة في كتابة البرامج عـن طريـق برنامـج (DBase III+) وكذلك باقي برامج عائلة (DBase) مثل (Dbase IV) ، (FoxBase +) ، (FoxPro) كبيرة جدا ولن يتسع المجال لدراستها بالتفصيل في هذا الكتاب . ولكن سيتم في هذا الباب القاء الضوء على معظم هذه الأوامر مع دراسة تفصيلية لها كلما أمكن . مع ملاحظة أن الأوامر مرتبة حسب الترتيب الهجائي للحروف الانجليزية . كما يستطيع القارىء الرجوع الى الملحق الموجود في نهاية هذا الكتاب لمعرفة الأوامر والدوال الاضافية لبرامج (DBase IV) ، (FoxBase +) ، (FoxPro) .

ملاحظة

القيم الموجودة داخل قوسين مربعين [ ] هي قيم اختيارية يستطيع المستخدم كتابتها او عدم كتابتها . كذلك فان مايكتب بين الزاويتين <> يعني عناصر مختارة بواسطة المستخدم نفسه . أما الزاويتين نفسهما فلا يكتبا ضمن العنصر .

١ - الأمر (?)

يستخدم هذا الأمر لعرض محتويات متغير ذاكرة أو حقل معين في ملف قاعدة البيانات عن طريق كتابة اسم هذا المتغير أو الحقل بعد علامة الاستفهام . وهو يعني استفهام أو سؤال عن قيمة مطلوبة .وتتم الكتابة من أول السطر التالي .

والصورة العامة للأمر كالآتي :

```
? [< expression list >]
```

حيث (expression List) هي اسم متغير ذاكرة أو حقل أو أي علاقة مطلوب حساب قيمتها .

وعند كتابة هذا الأمر بدون أي علاقة بعده فان هذا يؤدي الى عرض سطر خال . وتستخدم هذه العملية عندما يراد عرض سطور خالية بين المخرجات المطبوعة .

أمثلــة

لحساب نتيجة معادلة حسابية يتم كتابة السطر التالي :

```
? 2 * 2 + (8/2)
8
```

ولعرض تاريخ اليوم الحالى يتم كتابة السطر التالى :

```
. ? DATE()
01/07/90
```

ولعرض بيانات حقول فى قاعدة البيانات يتم كتابة السطور التالية :

```
. USE Cadets
. ? name , address
Mohamed Aly        12 - Nasr City - Cairo
```

٢ - الأمر (??)

هذا الأمر يؤدى نفس العمل مثل الأمر السابق تماما ، ولكنه يسمح بالكتابة من أى مكان فى السطر ولا يشترط الكتابة من أول السطر .

٣ - الأمر @

هذا الأمر يستخدم فى انشاء شاشات الادخال والتقارير بالاضافة الى التحكم فى مكان عرض البيانات على الشاشة .

وهناك عدة صور لهذا الأمر يتم دراسة كل منها على حدة . وهى :

أ - الصورة الأولى :

```
@ <row, col> SAY <expression list>
```

وتستخدم هذه الصورة فى عرض أى بيانات موجودة فى متغيرات ذاكرة ، أو حقول داخل ملف قاعدة البيانات ، أو رسائل خاصة يراد عرضها على المستخدم . ويتم ذلك عن طريق كتابة الاحداثيات المطلوب عرض البيانات بدءا منها . وهذه الاحداثيات هى رقم السطر (Row) ورقم العمود (Column) .

مـثـال

```
USE Cadets
@ 10,10 SAY name
```

وعند تنفيذ هذه الأوامر يظهر السطر التالى مثلا :

```
Ahmed Salem
```

وذلك بدءا من السطر العاشر والعمود العاشر .

ملاحـظة

الرقم المقابل للسطر (row) يأخذ أى قيمة من صفر الى ٢٣ ، والرقم المقابل للعمود (col) يأخذ أى قيمة من صفر الى ٧٩ . مع ملاحظة أن ترقيم السطور يبدأ من أعلى ، وترقيم الأعمدة يبدأ من اليسار . ويمكن تحويل السطور المكتوبة بواسطة هذا الأمر الى الطابعة عن طريق كتابة الأمر (SET DEVICE TO PRINT).

ب - الصورة الثانية

```
@ <row,col> SAY <expression list> PICTURE <clause>
```

وتستخدم هذه الصورة كالصورة السابقة تماما مع اضافة التحكم في شكل البيانات المعروضة عن طريق الأمر (PICTURE) . ويلى هذه الكلمة تعبير (Clause) يحدد صورة هذه البيانات . ويتم استخدام نموذج (Template) مكان هذا التعبير (Clause) . وهذا النموذج قد يكون مجموعة من الرموز (Symbols) يمثل كل منها حرفا من حروف البيانات التي تظهر في هذا السطر . وقد تكون دالة معينة (Function)يتم ادخالها للتحكم فى شكل جميع الحروف مرة واحدة .

رموز النموذج (Template Symbols)

وهى رموز يستخدم بعضها مع الأمر (@...SAY) ، ولكنها تستخدم فى الغالب مع الأمر (@...GET) كما سيتم الايضاح . والجدول التالى يوضح هذه الرموز ومعنى كل منها :

9 ويسمح بظهور الأعداد فقط .
ويسمح بظهور الأعداد وعلامات الجمع والطرح والمسافات بين الأرقام .
A ويسمح فقط بظهور الحروف الهجائية .
L ويسمح بظهور البيانات المنطقية .
Y ويسمح بظهور البيانات المنطقية Y , y , N , n .
N ويسمح بكتابة الحروف أو الأرقام .

X ويسمح بكتابة أى حروف أو حروف خاصة .
! وهو يحول الحروف الصغيرة الى كبيرة (Capital) .
$ وهو يعرض علامة الدولار قبل العدد .
. وهو يحدد مكان العلامة العشرية (Decimal Point) .

دوال النموذج (Template Functions)

كما سبق الايضاح فان هذه الدوال تتحكم فى شكل الحروف بالكامل . ولايلزم كتابة دالة لكل حرف . ويمكن الجمع بين الدالة (Function) والرموز (Symbols) فى نفس التعبير بعد الأمر (PICTURE) على أن يتم كتابة الدالة أولا ثم الرموز . وتبدأ الدالة عادة بالحرف @ . كما يمكن كتابة كلمة (FUNCTION) بدلا من الحرف @ حتى لايحدث خلط بينه وبين الأمر @ الموجود فى أول السطر .

والجدول التالى يوضح كل دالة ومعنى كل منها :

C وهى تعرض الحـــروف (CR) أى (Credit) بعد الأعداد السالبة .
(وهى تعرض الأعداد السالبة بين قوسين .
B وهى تؤدى الى ضبط الأعداد من اليسار .
Z وهى تؤدى الى حذف الأرقام التى قيمتها صفر .
D وهى تؤدى الى عرض التواريخ بالصورة الأمريكية .
E وهى تؤدى الى عرض التواريخ بالصورة الأوربية .
A وهى تؤدى الى عرض الحروف الهجائية فقط .
! وهى تؤدى الى تحويل الحروف الى حروف كبيرة (Capital).
R وهى تؤدى الى عرض حروف خاصة بين البيانات المعروضة .
S<n> وهى تؤدى الى تحديد عرض البيانات المعروضة بعدد (n) من الحروف . وتسمح بزحزحة الحروف (Scrolling) خلال هذا العرض .

أمثلـــة

السطر التالى يؤدى الى ضبط العدد الموجود فى الحقل (amount) جهة اليسار (Left Justified) وذلك عن طريق الدالة (B) .

```
@ 5,5 SAY amount PICTURE '@B 9,999,999.99'
```

كما يمكن كتابة نفس السطر السابق بصورة أخرى كالآتى :

```
@ 5,5 SAY amount FUNCTION 'B';
    PICTURE '9,999,999.99
```

ويمكن كتابة عدة دوال مع نفس الصورة (PICTURE) كالآتى :

```
@ 5,5 SAY amount PICTURE '@ XC 999.99'
```

جـ الصورة الثالثة

```
@<row,col> GET <variable> PICTURE <clause>
RANGE <exp>,<exp>
```

وتستخدم هذه الصورة فى عرض عمود ضوئى على الشاشة يمثل المتغير (variable) المكتوب . ومن خلال هذا العمود الضوئى يستطيع المستخدم ادخال البيانات المطلوبة والتى يتم تخزينها فى المتغير . ويجب ملاحظة أن هذا المتغير يلزم انشاؤه أولا قبل كتابة هذا الأمر .

وكلمة (PICTURE) تم شرحها فى الصورة السابقة ، وهى تؤدى هنا الى التحكم فى شكل البيانات التى يدخلها المستخدم . كما تؤدى الى تحويلها الى الشكل المطلوب ادخاله فى المتغير .

وكلمة (RANGE) تستخدم مع المدخلات العددية والتاريخية لتحديد أقل قيمة وأكبر قيمة مطلوب ادخالها فى المتغير .

ولتحديد مدى تاريخى مثلا يجب أولا تحويل التاريخ من الحروف الى تاريخ عن طريق الدالة (CTOD) . فمثلا يمكن كتابة المدى بين تاريخين كالآتى :

```
RANGE CTOD('01/01/90'), CTOD('02/05/90')
```

وهذا يحدد المدى من ١ /١ /٩٠ الى ٥ /٢ /٩٠

ويمكن كتابة الحد الأدني فقط أو الحد الأقصى فقط كالاتى مثلا: (Range 30,) . وهذا يعنى أي عدد يزيد عن (٣٠). أما (Range , 100) فهو يعنى أى عدد أقل من (١٠٠) .

د - الصورة الرابعة

```
@ <row1, col1> TO <row2, col2> [ DOUBLE ]
```

وتستخدم هذه الصورة فى رسم مستطيل يبدأ من النقطة الممثلة بالاحداثيات <row1، col1> التى تمثل أعلى نقطة يسار هذا المستطيل وينتهي بالنقطة الممثلة بالاحداثيات < row2, col2 > التى تمثل أدنى نقطة يمين المستطيل .

واستخدام (DOUBLE) اختيارى وهو يؤدى الى رسم المستطيل بخطوط مزدوجة .

وتفيد هذه الصورة فى رسم أشكال هندسية فى شاشة ادخال البيانات تعطى الشاشة شكلا مثيرا وجذابا .

أمثلة

- السطر التالى يؤدى الى رسم مستطيل بخطوط مفردة (Single) .

```
@ 2,20 TO 8,60
```

- السطر التالى يؤدى الى رسم مستطيل بخطوط مزدوجة (Double) .

```
@ 1,10 TO 7,50 DOUBLE
```

- السطر التالى يؤدى الى رسم خط أفقى

```
@ 3,5 TO 3,30
```

وذلك لأن رقم السطر ثابت فى النقطتين .

- السطر التالى يؤدى الى رسم خط رأسى

```
@ 1,20 TO 20,20
```

وذلك لأن رقم العمود ثابت فى النقطتين .

هـ ـ الصورة الخامسة

```
@ <row1, col1> CLEAR TO <row2, col2>
```

وتستخدم هذه الصورة فى مسح مستطيل من الشاشة يبـــدأ مـــن النقطـــة < row1, col1 > وينتهى بالنقطة < row2, col2> . ويمكن استخدام هذه الصورة فى مسح أجزاء مختلفة من الشاشة حسب الحاجة . كما يمكن مسح سطر واحد عن طريق كتابة أول نقطة فى هذا السطر وآخر نقطة فى السطر .

مـــثال

```
@ 1,0 CLEAR TO 6,50
```

وهذا الأمر يؤدى الى مسح مستطيل يبدأ من النقطة (0 , 1) وينتهى بالنقطة (50 , 6) .

٤ - الأمر (ACCEPT)

يستخدم هذا الأمر فى عرض رسالة للمستخدم واستقبال قيمة معينة يدخلها المستخدم رداً على هذه الرسالة . حيث يتم تخزين هذه القيمة فى متغير ذاكرة يتم انشاؤه من خلال هذا الأمر .

والصورة العامة لهذا الأمر كالآتى :

```
ACCEPT [<message>] TO <memvar>
```

ويجب ملاحظة أن الرسالة (message) في هذه الحالة يمكن أن تكون متغير ذاكرة حرفى (Character) أو تكون سلسلة حرفية يتم كتابتها بين علامات تنصيص (Quotation) .

كما يجب ملاحظة أن المتغير (memvar) يكون دائما متغيرا حرفيا (Character) .

مثال

```
ACCEPT "Enter your name:" TO mname
```

فى هذه الحالة تظهر الرسالة التالية للمستخدم :

```
Enter your name :
```

وعندما يكتب المستخدم اسمه ويضغط على مفتاح الادخال يتم تخزين هذا الاسم فى متغير الذاكرة (mname) .

٥ - الأمر (APPEND)

يستخدم هذا الأمر فى اضافة سجلات جديدة الى ملف قاعدة البيانات . وهو يستخدم شاشة الادخال المستخدمة سواء كانت شاشة الادخال المبدئية (Default) أو شاشة الادخال التى يتم تصميمها من خلال البرنامج .

والصورة العامة للأمر كالآتى :

```
APPEND [ BLANK ]
```

ويستخدم الاختيار (BLANK) لاضافة سجل خال فى نهاية ملف قاعدة البيانات حتى يمكن احلال بيانات الحقول مكان الحقول الخالية من خلال البرنامج .

مثال

يمكن كتابة الأوامر التالية :

```
.USE Cadets
.APPEND
```

وعند الضغط على مفتاح الادخال تظهر الشاشة كما فى المثال بشكل (٢٧ - ١)

NAME	TELEPHONE
ADDRESS	CLASS
NATIONALITY	HOBBIES
FATHER NAME	BIRTH DATE
FATHER JOB	RELIGION
MOTHER NAME	NOTES

شكل (٢٧ - ١)

وعند الانتهاء من ادخال بيانات الحقول يتم الضغط على مفتاح الادخال فيتم تخزين هذا السجل . كما يمكن الضغط على مفتاحي (Ctrl-End) لتخزين جميع السجلات التى تمت اضافتها والعودة الى مشيرة النقطة (Dot Prompt) .

٦ - الأمر (APPEND FROM)

يستخدم هذا الأمر في نسخ سجلات من ملف الى نهاية ملف قاعدة البيانات المفتوح . ولا يشترط أن يكون الملف المنسوخ منه ملف قاعدة بيانات كما سنلاحظ من الصور المختلفة للأمر . وانما يمكن أن يكون ملف جدول الكترونى (Spread Sheet) أو أى ملف آخر مكتوب بشفرة الآسكى (ASCII Code) .

وهناك صورتان للأمر ، الصورة الأولى تستخدم للنسخ من ملف قاعدة بيانات آخر، والصورة الثانية تستخدم للنسخ من ملفات أخرى .

أ - الصورة الأولى

```
APPEND FROM <Filename> [FOR <condition>]
```

حيث (filename) هو اسم الملف المنسوخ منه . ولايكتب فيه الامتداد (Extension) حيث أن البرنامج يضيف الامتداد (.dbf) آليا .

و (condition) هو الشرط الذى يحدد السجلات المطلوب نقلها ويجب ملاحظة أن الحقول المشتركة فى الملفين فقط هى التى يتم نسخ بياناتها . ولايشترط أن تكون بنفس الترتيب فى الملفين . واذا كان أى حقل فى الملف المنسوخ منه أكبر من الحقل المقابل فى الملف الآخر ، يتم حذف الحروف الزائدة (Truncation) اذا كان الحقل حرفيا ، كما يتم استبدال الأرقام بنجوم (Astriks) اذا كان الحقل عدديا .

مثال

نفرض أنه قد تم انشاء ملف قاعدة بيانات اسمه (School1) ويراد اضافة بيانات الطلبة الناجحين فقط فى المدرسة الى ملف آخر اسمه (Cadets) . لتنفيذ ذلك يتم كتابة الأوامر التالية :

```
USE Cadets
APPEND FROM School1 FOR grade>50
```

فى هذه الحالة يتم اضافة السجلات التى تحقق الشرط فقط الى الملف (Cadets) .

ب - الصورة الثانية

```
APPEND FROM <Filename> TYPE <Filetype>
```

وتستخدم هذه الصورة فى اضافة سجلات من ملفات ليست مكتوبة بواسطة (DBase III+) وهى تشمل الأنواع التالية :

١ - الملفات ذات الامتداد (SDF) وهى الملفات المكتوبة بشفرة الآسكى (ASCII Code) . ويتم نسخها حرفا حرفا بحيث ينتهى كل سجل بالكود الخاص بالادخال (Carriage Return) .

٢ - الملفات ذات الامتداد (SYLK) وهى الملفات المكتوبة بواسطة برنامج (VisiCalc) ، وهو برنامج جداول الكترونية . حيث يتم تحويل السطور (rows) الى سجلات والأعمدة (Columns) الى حقول .

٣ - الملفات ذات الامتداد (WKS) وهى الملفات المكتوبة بواسطة برنامج لوتس ١٢٣ (Lotus 123) حيث يتم تحويل السطور الى سجلات والأعمدة الى حقول .

ويجب ملاحظة كتابة الامتداد عند كتابة اسم الملف المنسوخ منه (Filename) ، وكذلك كتابة نوع الملف المنسوخ منه بعد كلمة (TYPE) وذلك كالآتى مثلا :

```
APPEND FROM ACCOUNTS.WKS TYPE WKS
```

٧ - الأمر (ASSIST)

ويستخدم هذا الأمر فى تشغيل القوائم الرئيسية لبرنامج المساعد (ASSISTANT) التى يمكن عن طريقها انشاء ملف قاعدة البيانات والملفات المرتبطة به مثل ملفات الفهرس (Index Files) ، وملفات التشكيل (Format Files) ، وملفات البحث (Query Files) ، وملفات المنظر (View Files) ، بالاضافة الى ملفات التقارير (Reports) والعناوين المختصرة (Labels) . وتسمح هذه القوائم كذلك بتعديل هذه الملفات واجراء عمليات التصحيح والبحث والعرض لأى بيانات مطلوبة .

ويتم تنفيذ أى اختيار من القوائم عن طريق تحريك مؤشر القائمة حتى يصل الى الاختيار المطلوب ثم الضغط على مفتاح الادخال . ويلاحظ عند الوصول الى الاختيار النهائى المطلوب تنفيذه ، ظهور الأمر المقابل له بالكامل على السطر أعلى عمود الحالة (Status Bar) والذى يسمى سطر الفعل (Action Line) . وهذا الأمر الذى يظهر هو نفس الأمر الذى يمكن للمستخدم كتابته من مشيرة النقطة (Dot Prompt) للحصول على نفس النتيجة . وتتيح هذه الطريقة للمستخدم التعرف على الشكل (Syntax) الخاص بأى أمر أثناء العمل من خلال قوائم برنامج المساعد (Assistant) . كما يمكن الحصول على أى معلومات اضافية على الشاشة . وذلك عن طريق عرض شاشة مساعدة (Help) توضح للمستخدم شكل الأمر (Syntax) بالاضافة الى شرح خصائص استخدام هذا الأمر .

ويمكن تشغيل قوائم برنامج المساعد (Assistant) عن طريق كتابة الأمر (ASSIST من مشيرة النقطة (Dot Prompt) . كما يمكن تشغيلها أيضا عن طريق الضغط على مفتاح (F2) . ويلاحظ فى هذه الحالة ظهور القائمة الرئيسية على الشاشة . انظر الشكل (٢٧ - ٢) .

```
Set Up  Create  Update Position Retrieve Organize Modify Tools
 Database file
 Format for Screen
 Query
 Catalog
 View
 Quit dBASE III PLUS

ASSIST          |<C:>|                          |Opt: 1/6
Move selection bar - ↑↓. Select - ◄─┘. Leave menu - ←→. Help
```

شكل (٢٧ - ٢)

وهذه القلئمة تحتوى على ثمانية اختيارات يمكن تلخيصها كالآتى :

أ - قائمة التجهيز (Set Up)

تستخدم هذه القائمة فى فتح ملف قاعدة البيانات والملفات المرتبطة به . كما تستخدم أيضا فى الخروج من البرنامج . وعند فتح أى ملف تظهر قائمة بوحدات

الأقراص المتاحة لاختيار وحدة الأقراص التى تحتوى على الملف المطلوب فتحه .

وعند فتح ملف قاعدة البيانات وبعد اختيار وحدة الأقراص والملف المطلوب ، يظهر سؤال عما اذا كان الملف مفهرسا (Indexed) أم لا . فاذا كان الملف مفهرسا يتم كتابة (Y) فتظهر قائمة بأسماء ملفات الفهرس ويتم اختيار ملفات الفهرس الخاصة بملف قاعدة البيانات المفتوح على ألا يزيد عدد ملفات الفهرس المفتوحة عن سبعة ملفات . وكل ملف يتم اختياره تظهر أمامه

علامة (◀) ، مع ملاحظة أن أول ملف يتم اختياره يصبح هو ملف الفهرس الرئيسى (Master) بصرف النظر عن ترتيب هذا الملف فى القائمة . ويتم تخزين هذه الاختيارات عن طريق الضغط على مفتاح السهم شمال (<--) .

وعند فتح ملف التشكيل (Format) يتم تحديد وحدة الأقراص فتظهر قائمة بملفات التشكيل الموجودة . ويتم اختيار ملف التشكيل المطلوب استخدامه .

ويمكن فتح ملف البحث (Query) وذلك باختيار ملف البحث المطلوب من قائمة الملفات التى تظهر على الشاشة .

كما يمكن فتح الكتالوج اذا كان قد سبق انشاء ملف كتالوج حيث يتم اختيار ملف الكتالوج المطلوب من قائمة الملفات التى تظهر على الشاشة . وفى هذه الحالة لاتظهر على الشاشة دائما الا الملفات المخزنة فى هذا الملف . فعند فتح ملف قاعدة البيانات أو الملفات المرتبطة به ، لاتظهر الا الملفات الموجودة فى ملف الكتالوج المفتوح . فى حين لو لم يتم فتح ملف الكتالوج تظهر جميع الملفات الموجودة على وحدة الأقراص المستخدمة (Current Drive) .

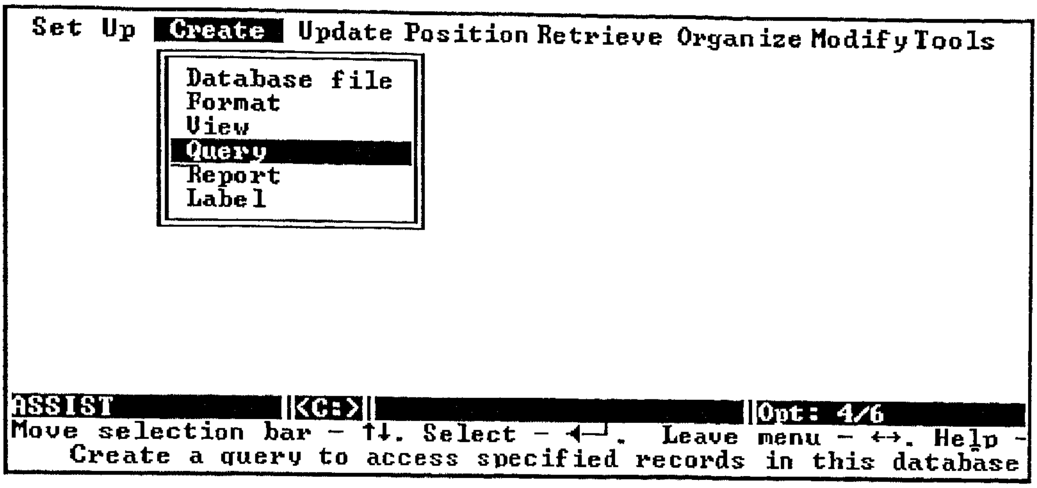
ويمكن فتح ملف المنظر (View File) بنفس الطريقة مثل الملفات السابقة . حيث يتم تحديد وحدة الأقراص المستخدمة ثم اختيار ملف المنظر من قائمة ملفات المنظر التى تظهر على الشاشة .

وللخروج من البرنامج يتم اختيار (Quit DBase III+) ويؤدى هذا الى اغلاق جميع الملفات المفتوحة والعودة الى نظام التشغيل .

ب - قائمة الانشاء (Create)

تستخدم هذه القائمة فى انشاء الملفات . ولكل نوع من الملفات يتم اختيار وحدة الأقراص المطلوب تخزين الملف فيها ، ثم يتم كتابة اسم الملف . واذا كان قد سبق فتح كتالوج ، فان هذا الملف الذى يتم انشاؤه يدخل فى الكتالوج .

انظر شكل (٢٧ - ٢)



شكل (٢٧ - ٢)

والاختيار (Database File) يستخدم فى انشاء ملف قاعدة بيانات .

ارجع الى الأمر (CREATE) .

والاختيار (Format) يستخدم فى انشاء ملف تشكيل (Format File) . وهو ملف يؤدى الى التحكم فى شكل شاشة الادخال التى يتم عن طريقها ادخال البيانات .

ارجع الى الأمر (CREATE SCREEN) .

والاختيار (View) يستخدم فى انشاء ملف المنظر (View File) وهو يسمح بالتعامل مع عدة ملفات قواعد بيانات فى نفس الوقت .

ارجع الى الأمر (CREATE VIEW)

والاختيار (Query) يستخدم فى انشاء ملف بحث (Query File) وهذا يساعد على ترشيح ملف قاعدة البيانات (Filtering) للحصول على البيانات المطلوبة .

ارجع الى الأمر (CREATE QUERY) .

والاختيار (Report) يستخدم فى انشاء ملف التقرير الذى يحتوى على الامتداد (.Frm) .

ارجع الى الأمر (CREATE REPORT) .

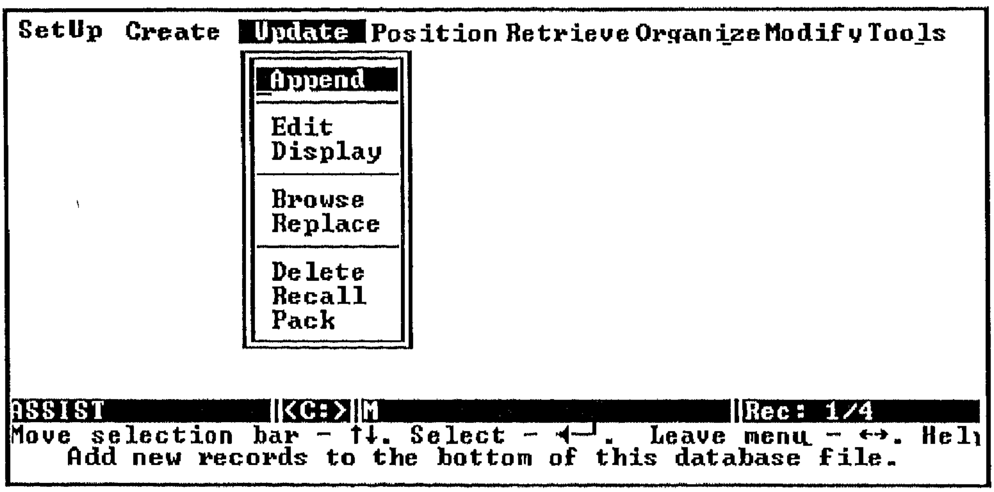
والاخــــتيار (Label) يستخدم فى انشاء ملف العناوين المختصرة (Label).

ارجع الى الأمر (CREATE LABEL) . .

جـ - قائمة التحديث (Update)

وتستخدم هذه القائمة فى تحديث البيانات المخزنة فى قاعدة البيانات . فيمكن من خلالها اضافة سجلات أو مسح سجلات أو تصحيح بيانات سجلات معينة فى الملف . وعند فتح ملفات الفهرس (Index Files) فان هذه الملفات يتم تحديثها آليا مع تحديث بيانات قاعدة البيانات .

انظر شكل (٢٧ - ٤)



شكل (٢٧ - ٤)

والاختيار (Append) يستخدم فى اضافة سجلات جديدة فى نهاية الملف .

ارجع الى الأمر (APPEND) .

والاختيار (Edit) يستخدم فى تعديل بيانات الملف سجلا سجلا ، وذلك حسب السجل الذى يقف عنده مؤشر الملف .

ارجع الى الأمر (EDIT) .

والاختيار (Display) يستخدم فى عرض بيانات سجل معين .

ارجع الى الأمر (DISPLAY) .

والاختيار (Browse) يستخدم فى تعديل بيانات الملف ، بالاضافة الى عرض حتى ١٧ سجلا على الشاشة .

ارجع الى الأمر (BROWSE) .

والاختيار (Replace) يسمح باجراء تعديلات مجمعة (Batch) لملف قاعدة البيانات . ويمكن استخدامه مثلا فى تعديل مرتب جميع الموظفين فى شركة معينة عن طريق ضرب حقل المرتب (Salary) فى نسبة ثابتة مثل (5 %) .

والاختيار (Delete) يستخدم فى وضع علامات (Marks) على السجلات المطلوب مسحها تمهيدا لمسحها تماما بواسطة الاختيار (Pack) .

والاختيار (Recall) يستخدم فى استعادة السجلات التى تم وضع علامات عليها لمسحها . ويمكن فى هذه الحالة استعادة بعض السجلات أو كلها حسب الحاجة .

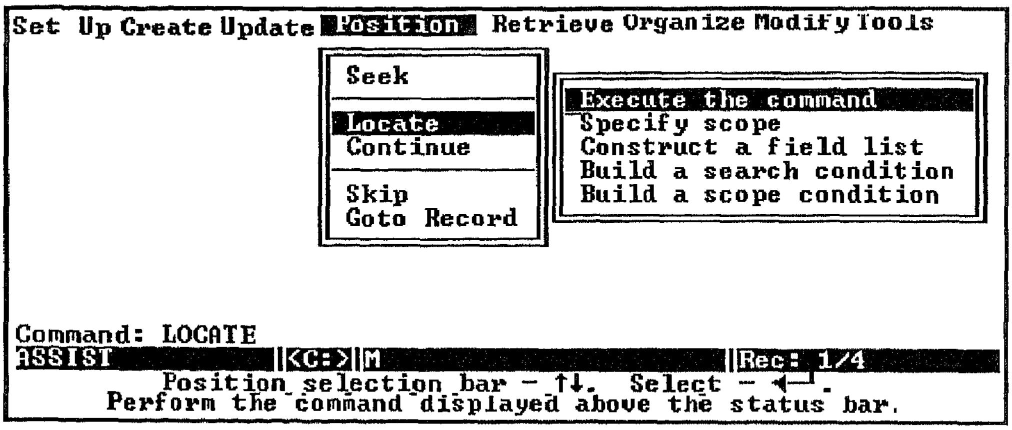
والاختيار (Pack) يستخدم فى المسح النهائى للسجلات التى تم وضع علامات عليها لمسحها .

د - قائمة المكان (Position)

وتستخدم هذه القائمة فى توجيه مؤشر السجلات (Record Pointer) الى سجل محدد ، وذلك حتى يمكن تحديث هذا السجل أو عرض بياناته . وفى كل اختيار من اختيارات هذه القائمة يقوم البرنامج بعرض القوائم الفرعية التى

يتم عن طريقها ادخال الشرط أو الشروط التى يتم بناء عليها توجيه المؤشر الى سجل محدد .

انظر شكل (٢٧ - ٥)



شكل (٢٧ - ٥)

والاختيار (Seek) يستخدم فقط عندما يكون الملف مفهرسا (Indexed) . وهو يوجه مؤشر السجلات الى أول سجل يحقق الشرط الذى يتم ادخاله .

ارجع الى الأمر (SEEK) .

والاختيار (Locate) يستخدم فى توجيه مؤشر السجلات (Record Pointer) الى أول سجل يحقق الشرط الذى يتم ادخاله .

ارجع الى الأمر (LOCATE) .

والاختيار (Continue) يستخدم فقط بعد استخدام الاختيار (Locate) . وذلك لعرض السجل الثانى الذى يحقق الشرط الذى سبق ادخاله .

والاختيار (Skip) يستخدم في تحريك مؤشر السجلات (Record Pointer) الى سجلات تالية أو سجلات سابقة حسب العدد الذى يتم ادخاله .

ارجع الى الأمر (SKIP) .

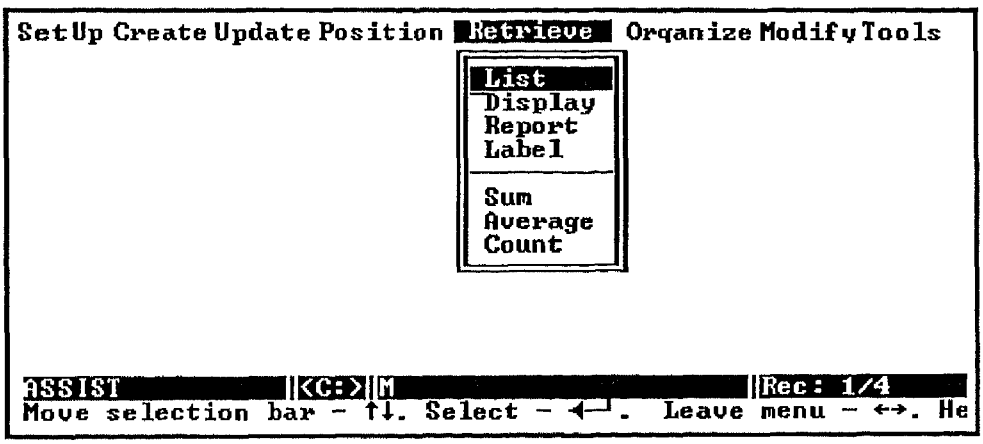
والاختيار (Goto Record) يستخدم فى تحديد رقم سجل معين يراد وضع المؤشر عنده .

ارجع الى الأمر (GOTO) .

هـ - قائمة الاسترجاع (Retrieve)

وتستخدم هذه القائمة فى عرض بيانات سجلات معينة على الشاشة أو طباعتها على الطابعة .

انظر الشكل (٢٧ - ٦)



شكل (٢٧ - ٦)

والاختيار (List) يستخدم لعرض بيانات جميع السجلات المطلوبة على الشاشة أو طباعتها على الطابعة حسب الحاجة .

ارجع الى الأمر (LIST) .

والاختيار (Display) يستخدم لنفس الهدف مثل الاختيار (List) ولكن هناك بعض الاختلافات بينهما .

ارجع الى الأمر (DISPLAY) .

والاختيار (Report) يستخدم فى عرض البيانات على الشاشة أو طباعتها على الطابعة حسب شكل التقرير (Report) الذى سبق انشاؤه .

ارجع الى الأمر (REPORT) .

والاختيار (Label) يستخدم فى عرض البيانات عل الشاشة أو طباعتها على الطابعة حسب شكل التقرير المختصر (Label) الذى سبق انشاؤه .

ارجع الى الأمر (LABEL) .

والاختيار (Sum) يستخدم لتجميع الحقول العددية لمجموعة من السجلات التى يتم اختيارها .

ارجع الى الأمر (SUM) .

والاختيار (Average) يستخدم لحساب القيم المتوسطة لكل الحقول العددية فى مجموعة من السجلات التى يتم اختيارها .

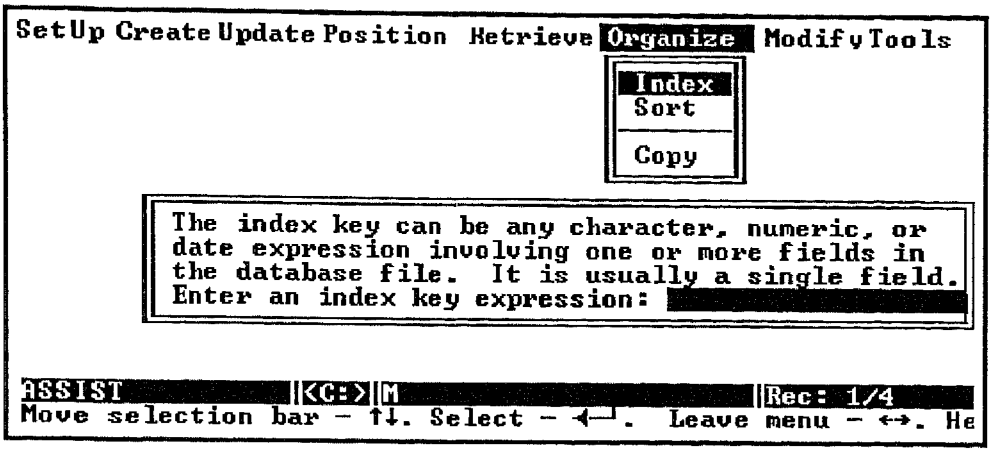
ارجع الى الأمر (AVERAGE) .

والاختيار (Count) يستخدم فى حساب عدد السجلات التى تحقق شرطا أو شروطا معينة .

ارجع الى الأمر (COUNT) .

و - قائمة التنظيم (Organize)

تستخدم هذه القائمة فى تنظيم السجلات وترتيبها داخل ملف قاعدة البيانات . وكل اختيار من اختيارات هذه القائمة يؤدى الى انشاء ملف جديد . انظر الشكل (٢٧ - ٧)



شكل (٢٧ - ٧)

والاختيار (Index) يؤدى الى انشاء ملف فهرسى (Index File) . وعند فتح هذا الملف من خلال قائمة التجهيز (Setup) ، فان سجلات ملف قاعدة البيانات تظهر دائما مرتبة حسب الحقل الفهرسى (Key Field) الذى تم اختياره وذلك رغم عدم حدوث أى تغيير فى الأماكن الفعلية للسجلات .

ارجع الى الأمر (INDEX) .

والاختيار (Sort) يؤدى الى انشاء ملف قاعدة بيانات جديد مرتب بالترتيب المطلوب .

ارجع الى الأمر (SORT) .

والاختيار (Copy) يستخدم فى عمل نسخة من ملف قاعدة البيانات . ويمكن نسخ الملف كله أو نسخ مجموعة من السجلات التى تحقق شرطا أو شروطا معينة فقط . كما يمكن أيضا نسخ بيانات مجموعة من الحقول فقط وليس كل الحقول .

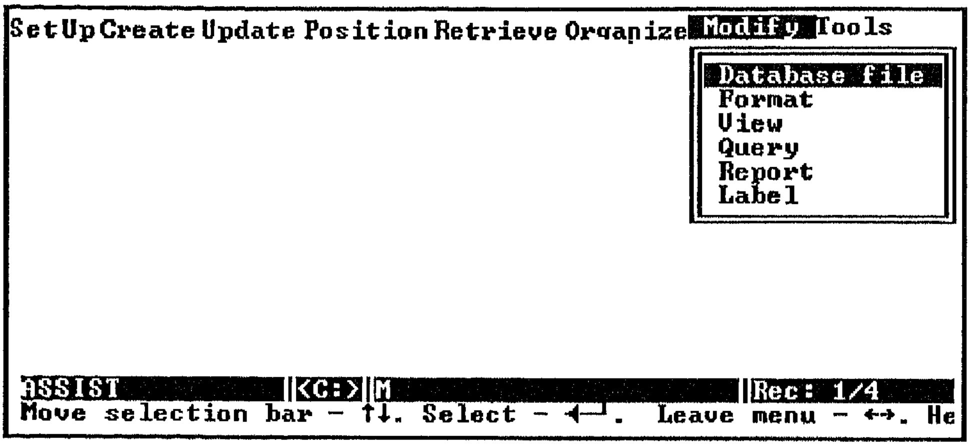
ارجع الى الأمر (COPY) .

ز - قائمة التعديل (Modify)

انظر الشكل (٢٧ - ٨)

وهذه القائمة تستخدم فى تعديل الملفات التى سبق انشاؤها من خلال قائمة الانشاء (Create) . وهى تحتوى على نفس الاختيارات الموجودة فى قائمة الانشاء (Create) ما عدا الكتالوج . وعند اختيار نوع الملف المطلوب تعديله سواء كان ملف قاعدة البيانات أو الملفات المرتبطة به ، فان البرنامج يعرض أسماء الملفات الموجودة على وحدة الأقراص من نفس النوع .

ارجع الى الأمر (MODIFY) .

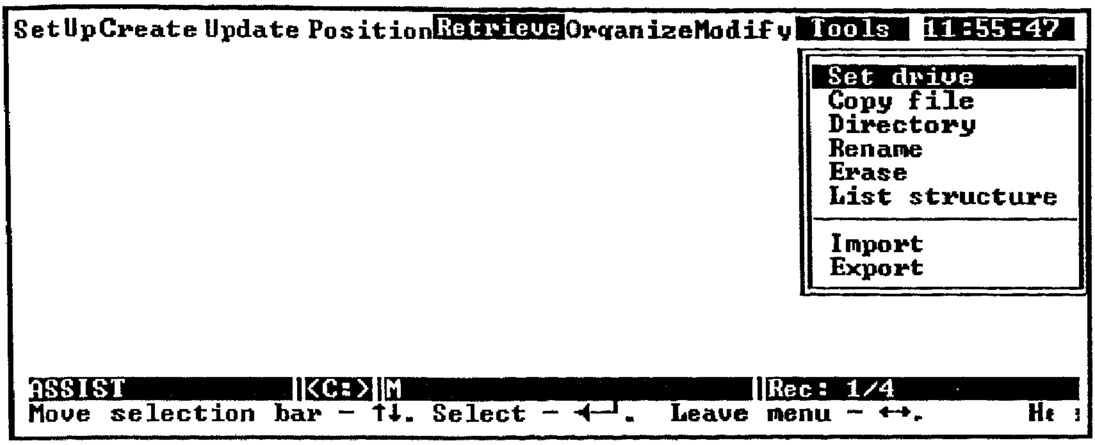


شكل (٢٧ - ٨)

حـ - قائمة الأدوات (Tools)

وهذه القائمة تؤدى بعض الوظائف التى يقوم بها نظام التشغيل مثل عرض فهرس الملفات (Directory) ونسخ الملف (Copy) وتغيير اسم ملف (Rename) . وهكذا .

انظر الشكل (٢٧ - ٩)



شكل (٢٧ - ٩)

والاختيار (Set Drive) يستخدم فى تحديد وحدة الأقراص المستخدمة وذلك بادخال رمز وحدة الأقراص (... ,A, B, C) التى تحتوى على الملفات المطلوب استخدامها .

والاختيار (Copy File) يستخدم فى نسخ أى نوع من الملفات . وهو لذلك يختلف عن الاختيار (Copy) فى قائمة التنظيم (Organize) . حيث أن الاختيار (Copy) يستخدم فى نسخ ملفات قاعدة البيانات فقط ، أى التى تحتوى على الامتداد (dbf.) .

ارجع الى الأمر (COPY FILE) .

والاختيار (Directory) يستخدم فى عرض قائمة أسماء الملفات الموجودة فى وحدة الأقراص المستخدمة .

ارجع الى الأمر (DIR) .

والاختيار (Rename) يستخدم لتغيير اسم أى ملف بشرط ألا يكون مفتوحا فى هذا الوقت .

ارجع الى الأمر (RENAME) .

والاختيار (Erase) يستخدم فى مسح أى ملف من القرص بشرط ألا يكون

مفتوحا فى هذا الوقت .

ارجع الى الأمر (ERASE) .

والاختيار (List Structure) يستخـــدم فى عرض تركيب الملف المفتوح . وهذا الاختيار يسمح أيضا بطباعة هذا التركيب (Structure) على الطابعة .

ارجع الى الأمر (LIST STRUCTURE) .

والاختيار (Import) يستخدم فى تحويل الملفات التى تمت كتابتها عن طريق برامج أخرى غير برنامج (DBase III+) الى ملفات يمكن استخدامها مع برنامج (DBase III+) .

ارجع الى الأمر (IMPORT) .

والاختيار (Export) يستخدم في تحويل الملفات المكتوبة بواسطة برنامج (DBase III+) الى ملفات يمكن استخدامها بواسطة برامج أخرى .

ارجع الى الأمر (EXPORT) .

٨ - الأمر (AVERAGE)

ويستخدم هذا الأمر في حساب القيم المتوسطة للحقول العددية لمجموعة من السجلات . والصورة العامة للأمر كالآتى :

```
AVERAGE[<expression list>][<scope>][WHILE <condition>]
   [FOR <condition>] [TO <memvar list>]
```

ويلاحظ هنا أن جميع الاختيارات اختيارية أى يمكن كتابتها أو عدم كتابتها . وعند عدم كتابة أى شيء بعد الأمر (AVERAGE) ، فان ذلك يؤدى الى حساب القيم المتوسطة لجميع الحقول العددية لجميع سجلات قاعدة البيانات المفتوحة ، وعرض هذه المتوسطات على الشاشة .

أما الاختيارات الموجودة فانها تحدد الحقول المطلوب حساب متوسطها، كما تحدد المدى (Scope) أى السجلات المطلوب البحث خلالها، كما تحدد الشروط المطلوب

البحث بناء عليها، ثم تحدد أسماء متغيرات الذاكرة التى يتم تخزين القيم المتوسطة فيها .

والاختيار الأول (expression list) يتم من خلاله كتابة أسماء الحقول المطلوب حساب متوسطها .

والاختيار الثاني (scope) يتم من خلاله تحديد مدى السجلات المطلوب البحث خلاله . حيث يمكن البحث خلال كل الملف (ALL) أو خلال السجلات التالية للسجل الذى يقف عنده المؤشر (Rest) و ... وهكذا .

والاختيار الثالث (WHILE <condition>) يؤدى الى البحث عن السجلات التى تحقق شرطا أو شروطا معينة .

والاختيار الرابع (FOR <condition>) يؤدى نفس العمل ، أى يبحث عن السجلات التي تحقق شرطا أو شروطا معينة . ولكن الاختيار (WHILE) أسرع فى الوصول الى السجلات التى تحقق الشروط .

والاختيار الخامس (TO<memvar list>) يؤدى الى انشاء متغيرات الذاكرة (memvar list) التي يتم فيها تخزين المتوسطات المحسوبة ، وذلك بنفس الترتيب الذى يتم به كتابة هذه المتغيرات .

مـثال

للحصول على متوسطات حقل ساعات العمل (w_hours) وحقل المرتب (salary) لكل ملف قاعدة البيانات يتم كتابة السطر التالى :

```
AVERAGE w_hours, salary TO avg_hr, avg-sal
```

فى هذه الحالة يحتوى المتغير (avg_hr) على متوسط ساعات العمل كما يحتوى المتغير (avg_sal) على متوسط المرتبات .

٩ - الأمر (BROWSE)

ويؤدى هذا الأمر الى عرض شاشة لتعديل بيانات السجلات وادخال سجلات جديدة . وهو يؤدى نفس العمل الذى يتم عند الدخول فى قائمة التحديث (Update) واختيار (Browse) . ولكن كتابته من مشيرة النقطة (Dot Prompt) أو من خلال البرنامج تعطى مرونة أكبر فى تحديد الحقول

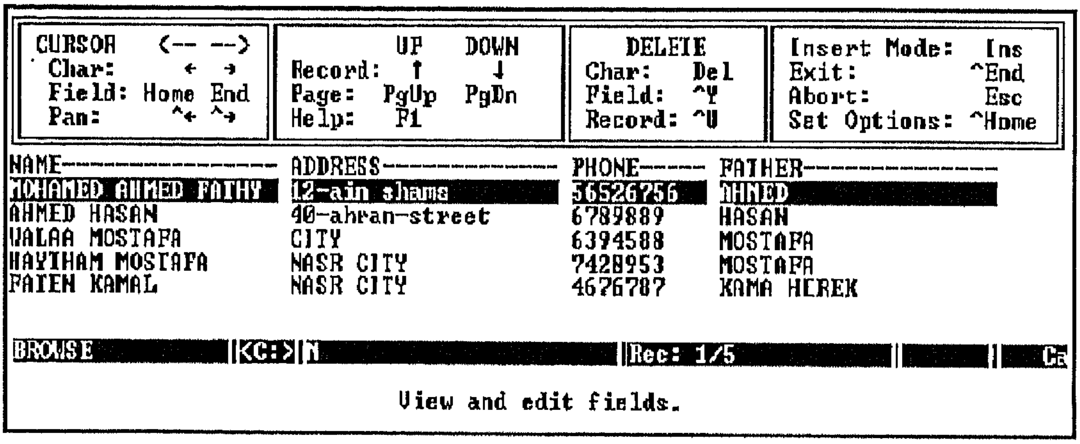
المطلوب عرضها على الشاشة .

والصورة العامة لهذا الأمر كالآتى :

```
BROWSE [FIELDS <field list>][LOCK <N>]
[FREEZE <field>][NOFOLLOW][NOMENU][WIDTH<N>][NOAPPEND]
```

ويلاحظ هنا أن جميع الاختيارات اختيارية ، أى يمكن كتابة الأمر (BROWSE) دون كتابة أى شيء بعده . وفى هذه الحالة تظهر شاشة تعديل البيانات وتحتوى على بيانات ١٧ سجلا . ويمكن عرض عدد أكبر من السجلات عن طريق الضغط على مفتاح (F1) لاخفاء قائمة المساعدة (Help) الموجودة أعلى الشاشة . وتظهر الحقول على هيئة أعمدة .

وعند الضغط على مفتاح (F10) يظهر عمود القوائم (menu bar) الخاص بالأمر (Browse) . وقد سبق شرح هذه القوائم فى الباب الخاص بتعديل البيانات فى الكتاب الأول . ويمكن ادخال معظم هذه الاختيارات مع الأمر (BROWSE) من خلال الاختيارات الموجودة مع الأمر والتى يمكن تلخيصها كالآتى : انظر الشكل (٢٧ - ١٠)



شكل (٢٧ - ١٠)

الاختيار (FIELDS <field list>) ويستخدم لتحديد أسماء الحقول المطلوب عرض البيانات الخاصة بها . ويمكن من خلال هذا الاختيار تحديد أى

ترتيب يراد عرض الحقول به . وذلك عن طريق كتابة أسماء الحقول (Field List) بنفس هذا الترتيب .

والاختيار (LOCK <N>) يستخدم لتثبيت عدد من الحقول المتجاورة في يسار الشاشة والتى لاتتحرك عند الضغط على مفتاحى (Ctrl,--->) أو (Ctrl,<---) لعمل زحزحة أفقية (Horizontal Scrolling) ويؤدى هذا الى عرض الحقول المختفية من الملف أمام مجموعة من الحقول تكون ظاهرة دائما على الشاشة . ويتم تحديد عدد الحقول المراد تثبيتها عن طريق العدد N .

والاختيار (FREEZE <field>) يستخدم لتجميد حقل معين حتى يصبح هو الحقل الوحيد المسموح بتعديله . وبالرغم من عرض باقى الحقول على الشاشة ، الا أن المستخدم لايستطيع تعديل أى حقل آخر غير هذا الحقل .

والاختيار (NOFOLLOW) يستخدم فقط مع الملفات المفهرسة (Indexed) . وهو يؤدى الى انتقال المؤشر الى السجل الجديد فى حالة تغيير الحقل الفهرسى (Key Field). وذلك لأن تغيير الحقل الفهرسى يؤدى الى تغيير ترتيب السجلات فى حين يظل المؤشر مكانه .

والاختيار (NOMENU) يستخدم لمنع استخدام عمود الاختيارات (Menu Bar) الخاص بالأمر (BROWSE).

والاختيار (WIDTH<N>) يستخدم لتحديد عرض الحقل المراد ظهوره بالحروف حيث يمثل العدد N عدد الحروف فى الحقل .

والاختيار (NOAPPEND) يستخدم لمنع المستخدم من اضافة أى سجلات جديدة ى الملف .

١٠ - الأمر (CALL)

ويتيح هذا الأمر لمخطط البرامج كتابة برامج منفصلة بلغة التجميع (Assembly Language) وتحميلها داخل ذاكرة الحاسب . وهذه البرامج يجب تحويلها أولا الى الشفرة الثنائية (Binary Code) . ويتم تحميلها فى الذاكرة باستخدام الأمر (LOAD) كما يمكن تنفيذها داخل البرنامج المكتوب بواسطة برنامج (DBase III+) باستخدام الأمر (CALL). حيث يتم كتابة هذا الأمر وبعده اسم البرنامج المطلوب تنفيذه . ويتم تحويل البرنامج من برنامج منفذ (Executable) الى برنامج ثنائى (Binary) عن طريق الأمر (EXE2BIN) وهو أمر من أوامر نظام التشغيل (MS-DOS) .

١١ - الأمر (CANCEL)

ويستخدم هذا الأمر لايقاف تنفيذ البرنامج والعودة الى مشيرة النقطة . والصورة العامة له كالآتى :

```
CANCEL
```

وهو لايحتاج الى معاملات أخرى .

مـثال

لايقاف تنفيذ البرنامج عند ضغط المستخدم على الحرف X يتم كتابـة الأوامـر التالية :

```
IF   Choice = 'X'
     CANCEL
ENDIF
```

١٢ - الأمر (CHANGE)

ويؤدى هذا الأمر الى عرض شاشة تصحيح مثل الشاشة التى تظهر مع استـخدام الاختيار (Edit) من قائمة التحديث (Update) . ولكن كتابة الأمر من مشيرة النقطة (Dot Prompt) أو من خلال البرنامج تتيح التحكم فى الحقول المطلوب تعديلها وكذلك تحديد السجل المطلوب تعديله بسهولة . والصـــورة العامة للأمر كالآتى :

```
CHANGE [<scope>] [FIELDS <field list>]
       [WHILE <condition>]  [FOR <condition>]
```

ويلاحظ أن جميع الاختيارات اختيارية ، أى يمكن كتابة الأمـر (CHANGE) دون كتابة أى شيء بعده . ويؤدى ذلك الى ظهور شاشة التصحيح التى يتم من خلالها تعديل السجلات على التوالى سجلا تلو الآخر بدءا من السجل الذى يقف عنده مؤشـر السجلات (Record Pointer) . ويتم الانتقال من السجل الى السجل التالى بالضغط على مفتاح (PgDn) .

أما الاختيارات الموجودة فانها تحدد الحقول المطلوب تعديلها وكذلك السجل المطلوب تعديله . ويمكن تلخيص هذه الاختيارات كالآتى :

الاختيار (<scope>) ويستخدم فى تحديد مدى السجلات المطلوب البحث خلالها . وقد يكون البحث خلال كل الملف (ALL) أو بدءا من السجل الذى يقف عند المؤشر وحتى آخر الملف (Rest) . كما يمكن تحديد سجل معين بكتابة رقمه .

والاختيار (FIELDS<field list>) يستخدم لتحديد أسماء الحقول المطلوب تعديلها . وهى الحقول التى تظهر على شاشة الادخال .

والاختيار (WHILE <condition) يستخدم للبحث عن السجلات التى تحقق الشرط أو الشروط التى يتم ادخالها كما يستخدم الاختيار (FOR <condition>) لنفس الغرض .

ملاحظة

الأمر (CHANGE) والأمر (EDIT) متماثلان تماما .

مثال

لتعديل حقول الاسم والعنوان وتاريخ الميلاد فى ملف بيانات الطلبة (Cadets) يمكن كتابة السطور التالية :

```
USE Cadets
CHANGE FIELDS name, address, birth_d
```

١٢ - الأمر (CLEAR)

ويستخدم هذا الأمر فى مسح الشاشة ووضع مؤشر الشاشة عند أعلى نقطة من اليسار ، وهى النقطة التى احداثياتها (صفر ، صفر) . كما يمكن مسح جزء فقط من الشاشة عن طريق كتابة الأمر (@....CLEAR) مع كتابة الاحداثيات المطلوب

المسح بدءا منها بعد الحرف @ . وذلك كالآتى مثلا :

```
@ 10,20 CLEAR
```

١٤ - الأمر (CLEAR ALL)

ويؤدي هذا الأمر الى اغلاق جميع ملفات قواعد البيانات المفتوحة والملفات المرتبطة بها مثل ملفات الفهرس (Index Files) وملفات التشكيل (Format Files) و ..... الخ . كما يؤدى هذا الأمر أيضا الى مسح متغيرات الذاكرة (Memory variables) .

١٥ - الأمر (CLEAR FIELDS)

ويستخدم هذا الأمر فى مسح الحقول التى سبق تحديدها بواسطة الأمر (SET FIELDS TO) . والأمر (SET FIELDS TO) يؤدى الى تحديد الحقول التى يتم استخدامها فقط من الملف ، فى حين تبقى باقي الحقول مغلقة وغير مستخدمة . ولذلك يستخدم الأمر (CLEAR FIELDS) فى اعادة الملف الى وضعه المبدئى . حيث تصبح جميع الحقول عاملة (Active) .

١٦ - الأمر (CLEAR GETS)

ويستخدم هذا الأمر فى مسح كل المتغيرات التى تم انشاؤها بواسطة الأمر (@...GET) . وهذه المتغيرات لاتزيد عن ١٢٨ متغيرا ، وذلك حسب الوضع المبدئى (Default) للبرنامج .

ويفضل دائما عند زيادة هذه المتغيرات أن يتم مسح بعضها بواسطة هذا الأمر ، وذلك بعد تخزينها فى متغيرات ذاكرة . خصوصا فى الحالات التى يراد فيها انشاء شاشات ادخال من عدة صفحات .

١٧ - الأمر (CLEAR MEMORY)

ويستخدم هذا الأمر في مسح متغيرات الذاكرة من الذاكرة المؤقتة للحاسب . وهو يماثل الأمر (RELEASE ALL) ، ولكن الاختلاف بين الأمرين أن الأمر (CLEAR MEMORY) يمسح كل متغيرات الذاكرة سواء كانت عامة (Public) أوخاصة(Private). أما الأمر (RELEASE ALL) فانه يمسح المتغيرات الخاصة فقط .

١٨ - الأمر (CLOSE)

ويستخدم هذا الأمر لاغلاق الملفات المفتوحة . وهو يكون على احدى صورتين :

```
CLOSE < file type >
```

أو

```
CLOSE ALL
```

والصورة الأولى يتم عن طريقها اغلاق نوع معين من الملفات . أما الصورة الثانية فانها تستخدم فى اغلاق جميع الملفات المفتوحة . ولتوضيح ذلك يمكن كتابة الأوامر التالية :

```
CLOSE DATABASES
CLOSE INDEX
CLOSE FORMAT
CLOSE PROCEDURE
CLOSE ALL
```

١٩ - الأمر (CONTINUE)

ويستخدم هذا الأمر مع الأمر (LOCATE) لتوجيه مؤشر السجلات (Record Pointer) الى السجل المطلوب . حيث يستخدم الأمر (CONTINUE) في تحريك المؤشر الى السجل الثانى الذى يحقق الشرط ثم السجل الذى يليه ، وهكذا .

ارجع الى الأمر (LOCATE) .

٢٠ - الأمر (COPY)

وهو يشبه الأمر (APPEND FROM) ولكنه لايضيف سجلات فى نهاية الملف المفتوح ، بل ينسخ الملف المفتوح كله أو جزءا منه فى ملف آخر جديد . وهناك صورتان لهذا الأمر . الصورة الأولى تستخدم فى نسخ ملف قاعدة البيانات كله أو جزء منه فى ملف قاعدة بيانات آخر . والصورة الثانية تستخدم فى نسخ ملف قاعدة البيانات كله أو جزء منه فى ملف آخر لايشترط أن يكون ملف قاعدة بيانات .

أ - الصورة الأولى

```
COPY TO <new filename>[<scope>][FIELDS <fieldlist>]
       [ WHILE <condition>] [FOR <condition>]
```

ويلاحظ أن جميع الاختيارات اختيارية ما عدا اسم الملف(<new filename>) .وعند كتابة الأمر بدون باقى الاختيارات كالآتى مثلا :

```
COPY TO <new filename>
```

فان هذا يؤدى الى نسخ الملف المفتوح بالكامل فى الملف الجديد الذى يتم كتابة اسمه .

أما اذا أريد نسخ حقول معينة أو سجلات معينة فقط فيتم استخدام الاختيارات الأخرى . وهذه الاختيارات تتلخص فى الآتى :

الاختيار (<scope>) ويستخدم لتحديد مدى محدد يراد البحث خلاله .

والاختيار (FIELDS<field list>) يستخدم لتحديد حقول معينة فقط يراد نسخها الى الملف الجديد .

والاختيار (WHILE<condition>) يستخدم فى البحث عن سجلات تحقق شرطا أو شروطا معينة لنسخها فى الملف الجديد . وكذلك يستخدم الاختيار(FOR<condition>) لنفس الغرض .

ب - الصورة الثانية

```
COPY TO <new filename>  [TYPE  <file type>]
```

وتستخدم هذه الصورة فى نسخ الملف المفتوح فى ملف آخر يمكن استخدامه بواسطة برامج اخرى غير برنامج (DBase III+) . ويجب في هذه الحالة اضافة الامتداد الخاص بهذا الملف . كما يجب كتابة نوع هذا الملف مكان الاختيار (file type) .

فمثلا لنسخ ملف قاعدة البيانات الى ملف يمكن استخدامه بواسطة برنامج لوتس ١-٢-٣ (Lotus 123) يتم كتابة السطر التالى :

```
COPY TO cadets.wks TYPE wks
```

كما يمكن نسخ مجموعة من الحقول فقط وكذلك مجموعة من السجلات عن طريق كتابة الاختيارات التى سبق شرحها فى الصورة الأولى .

٢١ - الأمر (COPY FILE)

يستخدم هذا الأمر لنسخ أي نوع من الملفات . والصورة العامة له كالآتى :

```
COPY FILE <file1> TO <file2>
```

ويجب ملاحظة أن أسماء الملفات هنا يجب أن تتضمن الامتداد (Extension) ووحدة الأقراص الموجود عليها كل ملف . كما يجب ملاحظة أن الملفات يجب الا تكون مفتوحة .

وعند نسخ ملف قاعدة بيانات (DBase file) يحتوى علي حقل ملاحظات (memo) فيجب نسخ ملف الملاحظات المرتبط به . وهو الملف الذى يحتوى على الامتداد (.dbt) .

٢٢ - الأمر (COPY STRUCTURE)

يستخدم هذا الأمر في نسخ تركيب ملف قاعدة البيانات فقط دون نسخ السجلات المخزنة به . والصورة العامة له كالآتى :

```
COPY STRUCTURE TO <filename>  [FIELDS <field list>]
```

واسم الملف يجب أن يشمل وحدة الأقراص الموجود عليها الملف اذا كانت غير وحدة الأقراص المبدئية (Default) . ويستخدم الاختيار (FIELDS) في تحديد حقول معينة يراد نسخها في هذا التركيب .

٢٣ - الأمر (COPY STRUCTURE EXTENDED)

يستخدم هذا الأمر في نسخ تركيب ملف قاعدة البيانات في ملف آخر يحتوى على أربعة حقول فقط وهى (Field Name) ، (Field Type) ، (Field Length) ،(Decimal numbers) .

ويتم ادخال حقول الملف المنسوخ الى الملف الجديد كسجلات . وتستخدم هذه الطريقة في تصميم البرامج التطبيقية التى يراد السماح للمستخدم بتعديل تركيب ملف قاعدة البيانات من خلالها . دون الحاجة الى استخدام الأمر (MODIFY STRUCTURE) . حيث يتم تعديل تركيب الملف مثل تعديل أى سجل في قاعدة البيانات . والصورة العامة للأمر كالآتى :

```
COPY TO <new file> STRUCTURE EXTENDED
```

وبعد السماح للمستخدم بادخال التعديلات المطلوبة على هذا التركيب ، يتم انشاء ملف قاعدة بيانات جديد من هذا التركيب باستخدام الأمر (CREATE FROM) . فمثلا اذا كان هناك ملف للموظفين يحتوى على التركيب التالى:

Field	Field Name	Type	width	Dec
1	name	character	30	
2	address	character	30	
3	phone	character	10	

فعند استخدام الأمر التالى :

```
COPY TO newname STRUCTURE EXTENDED
```

يصبح تركيب الملف كالآتى :

FFIELD	Field Name	Type	width	Dec
1	Field_name	character	10	
2	Field_type	character	1	
3	Field_len	Numeric	3	
4	Field_dec	Numeric	3	

وعند عرض سجلات هذا الملف بواسطة الأمر (List) مثلا يظهر الآتى :

Record	Field_Name	Field_Type	Width-Len	Field-dec
1	name	C	30	
2	address	C	30	
3	phone	C	10	

وفي هذه الحالة يمكن تعديل حقول الملف باستخدام أى أمر من أوامر التعديل مثل (Edit) ، (Change) .

٢٤ - الأمر (COUNT)

ويستخدم هذا الأمر فى حساب عدد السجلات التى تحقق شرطا أو شروطا معينة . والصورة العامة للأمر كالآتى :

```
COUNT[<Scope>][WHILE<codition>]
    [FOR <condition>][TO <memvar>]
```

ويلاحظ أن جميع الاختيارات اختيارية ، حيث يمكن كتابة الأمر (COUNT) دون كتابة أى شيء بعده . وفى هذه الحالة يتم حساب عدد السجلات فى ملف قاعدة البيانات المفتوح .

أما اذا أريد حساب عدد السجلات التى تحقق شروطا معينة ، يتم استخدام الشروط الموجودة مع الأمر . كما يمكن استخدام متغير الذاكرة (memvar) فى تخزين هذا العدد لاستخدامه فى البرنامج حسب الحاجة .

فمثلا يمكن كتابة الأمر التالى :

```
COUNT FOR name = "Mohamed" TO mname
```

فى هذه الحالة يتم حساب عدد الأشخاص الذين يبدأ اسمهم بالاسم (Mohamed) ثم يتم تخزين هذا العدد فى المتغير (mname) .

٢٥ - الأمر (CREATE)

يستخدم هذا الأمر فى انشاء ملف قاعدة البيانات . وهو يؤدى الى ظهور الشاشة التى تظهر عند استخدام قائمة الانشاء (Create) فى القائمة الرئيسية لبرنامج المساعد (Assistant) واختيار (Database file) .

ارجع الى الجزء الخاص ببرنامج المساعد (Assistant) فى الكتاب الاول .

٢٦ - الأمر (CREATE FROM)

ويستخدم هذا الأمر فى انشاء ملف قاعدة بيانات من ملف سبق نسخه بواسطة الأمر (COPY STRUCTURE EXTENDED) .

والصورة العامة له كالآتى :

```
CREATE <new file> FROM <structure extended file >
```

ارجع الى الأمر (COPY STRUCTURE EXTENDED) .

٢٧ - الأمر (CREATE/MODIFY LABEL)

يستخدم هذا الأمر فى انشاء العناوين المختصرة (Labels) ، وهى صورة مصغرة من التقارير (Reports) تعطى معلومات سريعة عن سجلات قاعدة البيانات . والصورة العامة للأمر كالآتى :

```
CREATE/MODIFY LABEL <filename>/?
```

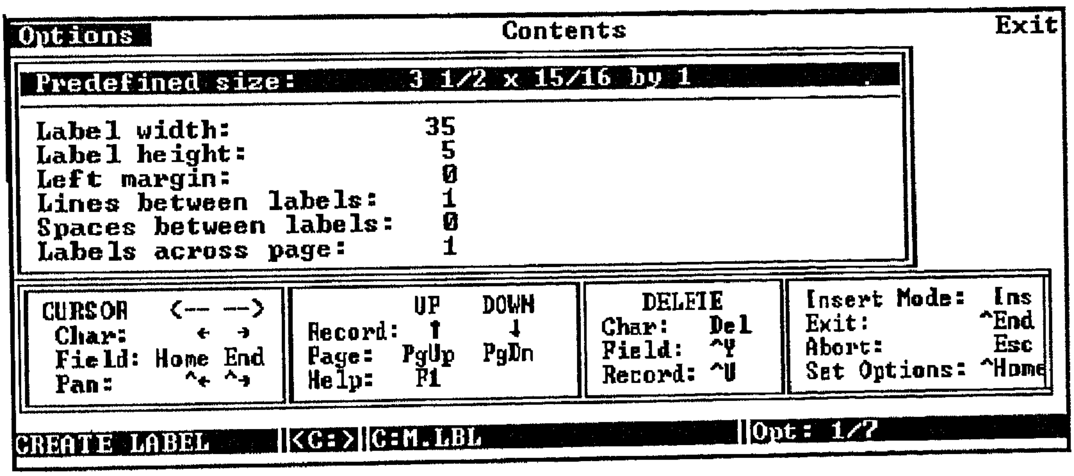
وهذا الأمر يؤدى الى ظهور نفس الشاشة التى تظهر عند الدخول فى قائمة الانشاء (Create) أو قائمة التعديل (Modify) واختيار (Label) ، مع ملاحظة

أن الأمر (CREATE) يستخدم فى انشاء التقرير والأمر (MODIFY) يستخدم فى انشائه أو تعديله بعد ذلك . ويتم كتابة اسم الملف بدون الامتداد ، حيث أن البرنامج يضيف الامتداد (lbl.) اليه آليا . واذا لم يتذكر المستخدم اسم الملف المطلوب فانه يكتب(?) بدلا من اسم الملف . وفي هذه الحالة تظهر قائمة بأسماء ملفات التقارير المختصرة (Labels) الموجودة فى القرص أو فى الكتالوج المفتوح .

ويحتوى عمود الاختيارات (Menu bar) الذى يظهر عند كتابة هذا الأمر على ثلاثة قوائم يمكن تلخيصها كالآتى :

١- قائمة الاختيارات (Options)

وهى تحدد حجم التقرير (Size) كما هو موضح في شكل (٢٧ - ١١) ، وهناك خمسة أحجام قياسية يظهر أحدها على أول سطر فى القائمة وتظهر باقى الأحجام عند الضغط على مفتاح الادخال . كما تستخدم باقى اختيارات القائمة فى تحديد عرض التقرير وارتفاعه والهامش الأيسر والمسافة بين التقارير ... وهكذا .



الشكل (٢٧ - ١١)

٢ - قائمة المحتويات (Contents)

ويتم عن طريقها تحديد محتويات التقرير ، أى أسماء الحقول أو متغيرات الذاكرة المثلة لها . وعند كتابة أسماء الحقول يمكن الضغط على مفتاح (F10) لعرض قائمة بأسماء الحقول واختيار الحقول المراد عرضها فى التقرير . ويمكن عرض أى عدد من السطور فى التقرير يحتوى كل منها على حقل أو أكثر . وعندما يراد عرض أكثر من حقل فى نفس السطر يتم فصلها بواسطة الفاصلة (,) .

٣ - قائمة الخروج (Exit)

وتستخدم هذه القائمة فى الخروج من قوائم التقارير المختصرة (Label) وهى تحتوى على اختيارين :

أ - الاختيار (Save) ويستخدم في تخزين التقرير الذى تم انشاؤه أو التعديلات التى تم ادخالها على تقرير سابق . ويمكن تنفيذ هذه العملية أيضا بالضغط على مفتاحى (Ctrl-End) .

ب- الاختيار (Abandon) ويستخدم فى الخروج من قوائم التقارير المختصرة دون تخزين التقرير أو التعديلات التى تم ادخالها على تقرير سابق .

٢٨ - الأمر (CREATE/MODIFY QUERY)

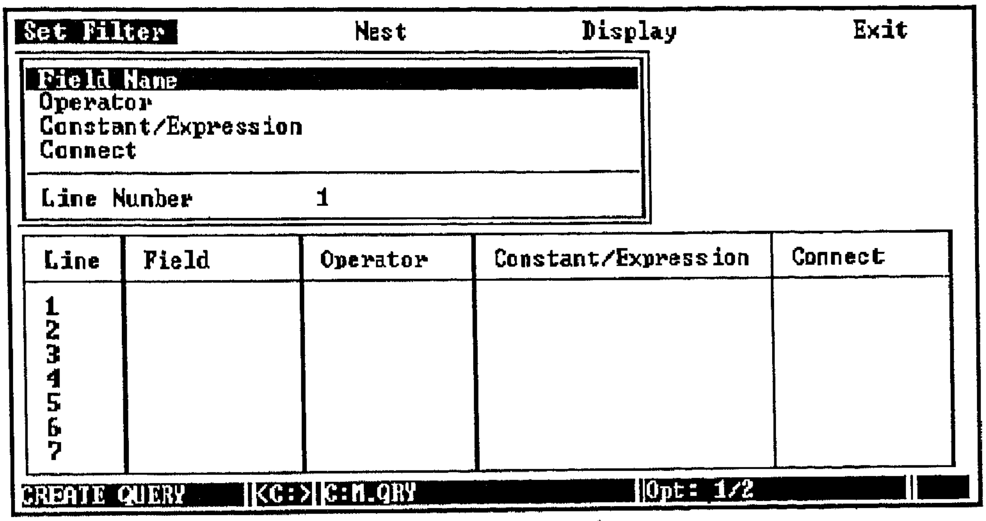
يستخدم هذا الأمر فى انشاء أو تعديل مرشح (Filter) لاستخدامه مع قاعدة البيانات . وهذا المرشح يؤدى نفس عمل المرشح بمفهومه الميكانيكى . حيث أنه لايسمح بالمرور الا للسجلات التى تحقق الشرط أو الشروط المطلوبة . والصورة العامة للأمر كالآتى :

```
CREATE/MODIFY QUERY <filename>/?
```

ويستخدم الأمر (CREATE) لانشاء ملف مرشح جديد . كما يستخدم الأمر (MODIFY) لتعديل ملف سبق انشاؤه . ويتم كتابة اسم الملف بدون الامتداد حيث أن البرنامج يضيفه آليا . واذا لم يتذكر المستخدم اسم الملف المطلوب يمكنه كتابة الحرف (?) لعرض أسماء ملفات البحث المخزنة على القرص أو فى الكتالوج لاختيار الملف المطلوب تعديله .

وهذا الأمر يؤدى الى ظهور نفس الشاشة التى تظهر عند الدخول فى قائمة الانشاء (Create) أو قائمة التعديل (Modify) واختيار (Query) . حيث يظهر عمود الاختيارات (Menu bar) الذى يحتوى على القوائم التالية :

انظر شكل (٢٧ - ١٢)



شكل (٢٧ - ١٢)

١ - القائمة (Set Filter)

وهذه القائمة تستخدم فى تحديد الشروط المطلوب ادخالها فى ملف البحث حيث تظهر عدة اختيارات تساعد المستخدم على ادخال هذه الشروط فى الجدول الذى يظهر على الشاشة ، وهذه الاختيارات تكون كالآتى :

- الاختيار الأول هو اسم الحقل (Field Name) الذى يؤدى الى عرض قائمة بأسماء الحقول لاختيار الحقل المطلوب ادخاله فى الشرط .

- الاختيار الثانى هو معامل المقارنة (Operator) ويؤدى الى ظهور قائمة بالمعاملات التى يتم اختيار معامل المقارنة المطلوب منها .

- الاختيار الثالث هو القيمة المطلوب مقارنتها (Constant/Expression) . وهو يتيح للمستخدم كتابة القيمة المطلوب مقارنة محتويات الحقل الذى تم اختياره بها .

- الاختيار الرابع هو الربط (Connect) وهو يساعد المستخدم على الربط بين عدة شروط باستخدام المعاملات المنطقية مثل (AND) ، (OR) .

- الاختيار الخامس هو رقم السطر (Line Number) وهو يساعد المستخدم على اختيار رقم سطر معين فى الجدول لتعديله . ويتغير هذا الرقم آليا عند الانتقال من سطر الى آخر .

ويمكن للمستخدم اضافة أي سطر بين السطور التى تمت كتابتها فى الجدول بالضغط على مفتاحى (Ctrl-N) . كما يمكنه أيضا مسح أى سطر من السطور بالضغط على مفتاحى (Ctrl-U) .

٢ - القائمة (Nest)

هذه القائمة تستخدم في وضع الأقواس حول الشروط التي يراد وضعها داخل الأقواس . وذلك للتحكم فى ترتيب تنفيذ الشروط المختلفة . وهى تتيح للمستخدم اضافة أقواس أو حذف أقواس سبق اضافتها فى أي سطر من سطورالجدول . ويفيد ذلك فى تكوين العلاقات المركبة التى تتكون من عدة شروط .

٣ - القائمة (Display)

وتستخدم هذه القائمة فى عرض السجلات التى تحقق الشروط التى سبق ادخالها . وذلك للتأكد أن الشروط قد تمت كتابتها بدقة . حيث يتم عرض بيانات أول سجل يحقق الشرط . ثم يتم الانتقال الى السجلات التالية عن طريق الضغط على مفتاح (PgDn) .

٤ - قائمة الخروج (EXIT)

وهى تستخدم لتخزين الملف الذى تم تكوينه أو الخروج دون تخزين الملف . ارجع الى الجزء الخاص بانشاء ملفات البحث عن طريق برنامج المساعد (Assistant) .

٢٩ - الأمر (CREATE /MODIFY REPORT)

يستخدم هذا الأمر في انشاء التقارير (Reports) الجديدة أو تعديل تقارير سبق انشاؤها . وهذه التقارير يتم تصميمها على هيئة أعمدة تمثل الحقول المختلفة . والصورة العامة للأمر كالآتى :

CREATE/MODIFY REPORT <filename>/?

وهذا الأمر يؤدى الى ظهور نفس الشاشة التى تظهر عند الدخول فى قائمة الانشاء (CREATE) واختيار (REPORT) . ويستخدم الأمر (CREATE) لانشاء ملف تقارير جديد فى حين يستخدم الأمر (MODIFY) فى انشاء أو تعديل ملف تقرير سبق انشاؤه . ويتم ادخال اسم الملف بدون الامتداد حيث أن البرنامج يضيف الامتداد (FRM.) آليا . واذا لم يتذكر المستخدم اسم الملف المطلوب تعديله فانه يكتب الحرف (?) لعرض جميع الملفات المخزنة على القرص أو فى الكتالوج المفتوح ، ثم اختيار الملف المطلوب .

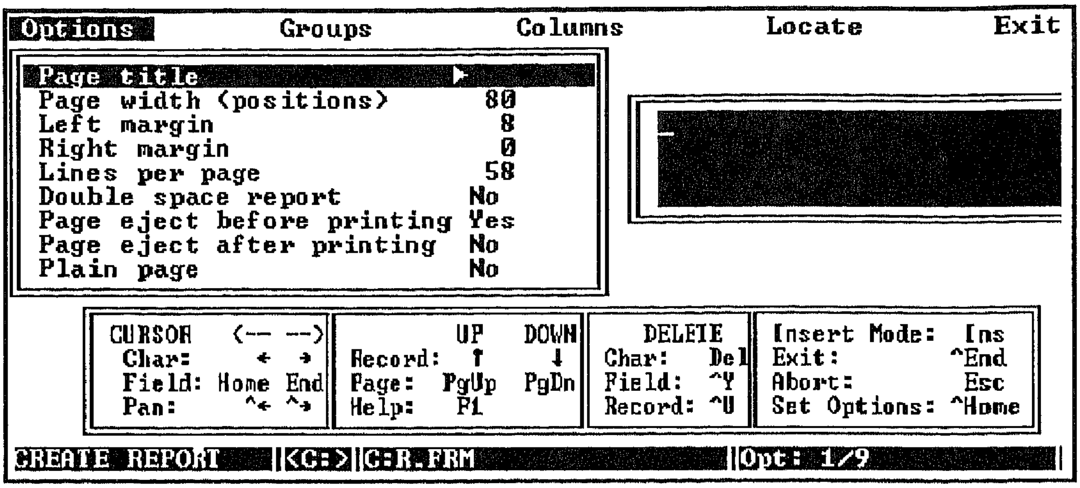
وعند كتابة هذا الأمر يظهر عمود الاختيارات (Menu Bar) الذى يحتوى على مجموعة من الاختيارات يمكن تلخيصها كالآتى :

أ- الاختيارات (Options)

وهي تحتوى على عدة اختيارات يتم عن طريقها تحديد عنوان للصفحة (Page Title) بالاضافة الى الاختيارات الخاصة بأبعاد الصفحة مثل العرض والهوامش وعدد السطور فى الصفحة و ... الخ .

وعند ادخال عنوان الصفحة (Page Title) ، فان البرنامج يتيح للمستخدم ادخال حتى (٤) سطور بحد أقصى (٦٠) حرفا . أما باقى الاختيارات فى القائمة فيمكن تعديلها حسب الحاجة ولكن فى الغالب تكون القيم المبدئية(Default) المكتوبة أمامها مناسبة .

انظر الشكل (٢٧ - ١٢)



شكل (٢٧ - ١٢)

ب - المجموعات (Groups)

استخدام المجموعات اختيارى ، وبعض التقارير لاتحتاج الى استخدام هذا الاختيار . ولكنها فى أحيان أخرى تكون مطلوبة لتحسين شكل التقرير والحصول على معلومات أكثر تفصيلا . حيث أن هذه المجموعات تتيح للمستخدم تقسيم بيانات التقرير بناء على البيانات الموجودة فى أحد الحقول . فمثلا اذا كانت هناك قاعدة بيانات للموظفين الذين يختلفون فى الوظائف (Jobs) . فيمكن تصميم التقرير مع استخدام حقل الوظيفة (Job) كأساس للتقسيم ، وذلك عن طريق كتابة اسم هذا الحقل مكان الاختيار (Group on Expression) وعند طباعة التقرير تظهر بيانات الموظفين الذين يشتركون فى الوظيفة فى مجموعة واحدة .

واذا كانت هناك حقول عددية فى ملف قاعدة البيانات ، فيتم تجميع هذه البيانات لكل مجموعة على حدة .

وهناك عدة اختيارات تحت قائمة المجموعات وهى كالآتى :

- الاختيار (Group on expression) وهو يستخدم لادخال اسم الحقل المراد التقسيم بناء عليه .

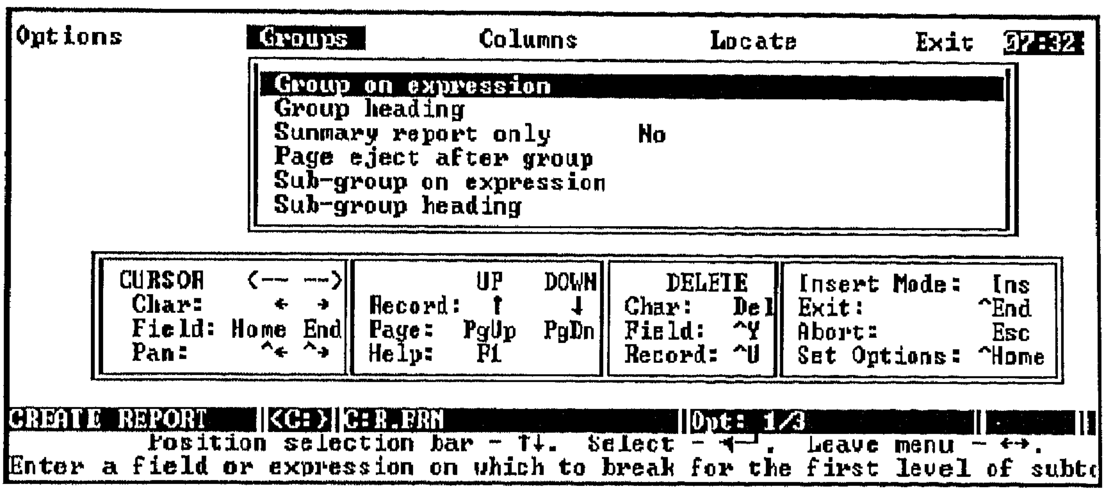
- الاختيار (Group Heading) ويستخدم لادخال عنوان لكل مجموعة يوضح الحقل الذى تم التقسيم بناء عليه . فمثلا يمكن كتابة (Client's Job) كعنوان لكل مجموعة .

- الاختيار (Summary Report Only) وهو يستخدم لتحديد ما اذا كان المطلوب الحصول على تقرير مختصر أو تقرير تفصيلي . وفى حالة التقرير المختصر يتم عرض البيانات العددية لكل مجموعة بعد تجميعها دون عرض باقى البيانات . أما التقرير التفصيلى فيتضمن جميع بيانات السجلات الموجودة فى كل مجموعة .

- الاختيار (Page Eject After Group) ويستخدم لكتابة بيانات كل مجموعة فى صفحة منفصلة .

- الاختيار (Sub-group on Expression) ويستخدم لتجميع البيانات فى مجموعات فرعية داخل المجموعات . ويتم ذلك بناء على حقل آخر يتم اختياره .

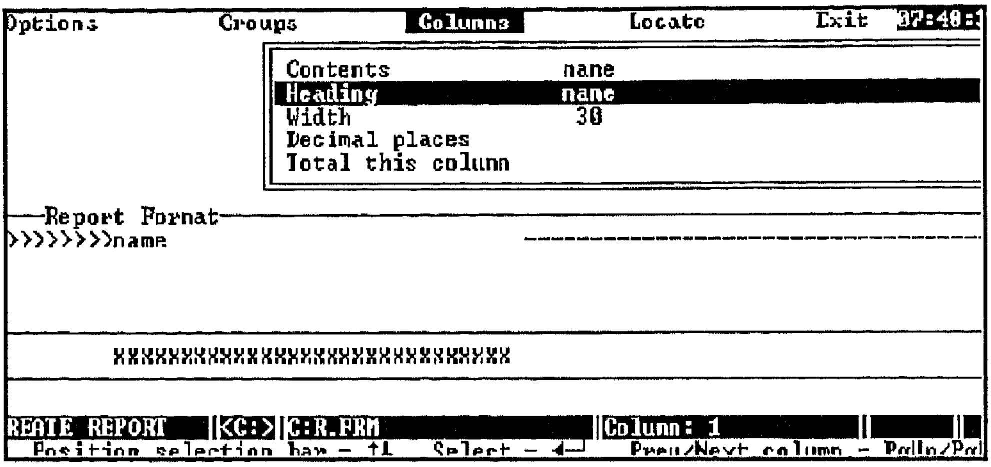
- الاختيار (Sub-group Heading) ويستخدم لتحديد عنوان لكل مجموعة فرعية . انظر شكل (٢٧ - ١٤)



شكل (٢٧ - ١٤)

جـ - الأعمـدة (Columns)

وهذه القائمة تستخدم فى تحديد البيانات المطلوب كتابتها فى الأعمـدة الخاصة بالتقرير ، مع تحديد مكان هذه البيانات وشكلها فى التقرير . وتحتوى هذه القائمة كما هو موضح فى شكل (٢٧ - ١٥) على الاختيارات التالية :



شكل (٢٧ - ١٥)

- الاختيار (Contents) ويستخدم فى تحديد محتويات كل عمود من أعمدة التقرير . ويمكن أن يحتوى العمود على حقل بيانات (Field) أو متغير ذاكرة (Memory Variable) أو علاقة رياضية بين أكثر من حقل .

- الاختيار (Heading) ويستخدم هذا الاختيار لكتابة عنوان للعمود .

- الاختيار (Width) ويستخدم هذا الاختيار لتحديد عرض العمود . وهو يكون نفس عرض الحقل الذى يتم ادخاله فى هذا العمـود . واذا كان العنوان (Heading) أطول من عرض الحقل ، فان عرض العمود يمتد ليغطى عرض العنوان . كما يمكن زيادة أو انقاص عرض العمود بكتابة العرض المطلوب . وعند كتابة عرض أقل من عرض الحقل ، فان البيانات التى تظهر فى هذا العمود ينتقل جزء منها الى السطر التالى .

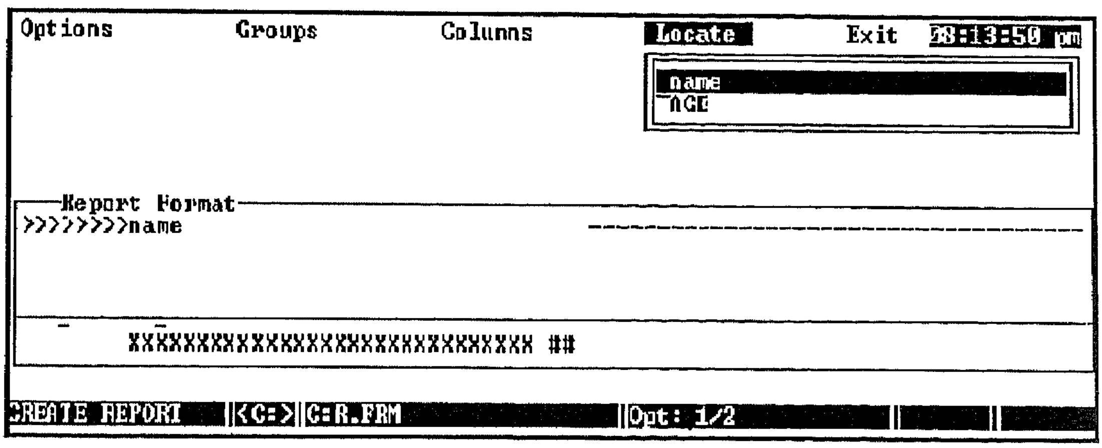
- الاختيار (Decimal Places) ويستخدم مع الحقول العددية فقط . وهو يحدد عدد الكسور العشرية المطلوب ظهورها فى التقرير لهذا العدد، وهو يأخذ نفس القيمة التى ســبق ادخالها عند انشاء ملف قاعدة البيانات . كمايمكن ادخال أى قيمة أخرى .

- الاختيار (Total This Column) ويستخدم مع الحقول العددية عندما يراد تجميع البيانات العددية الخاصة بهذا الحقل لمجموعة السجلات التى تظهر فى التقرير . والقيمة المبدئية لهذا الاختيار هى (Yes) ، واذا أريد تغييرها يتم كتابة (N) أمام هذا الاختيار . واذا أريد مسح العمود بعد ادخاله يتم الضغط على مفتاحى (Ctrl-U) . واذا أريد اضافة عمود قبل العمود الجارى ادخاله يتم الضغط على مفتاحى (Ctrl-N) .

د- الاختيار : (Locate)

يستخدم للوصول الى أى عمود سبق ادخال بياناته . حيث تظهر قائمة بأسماء الأعمدة التى سبق ادخالها ، فيتم اختيار العمود المطلوب تعديله .

انظر الشكل (٢٧ - ١٦)



شكل (٢٧ - ١٦)

هـ - الاختيار : (Exit)

ويستخدم لتخزين التقرير الذى تم انشاؤه أو التعديلات التى تم ادخالها . أو الخروج دون تخزين التقرير .

ارجع الى الجزء الخاص بانشاء التقرير فى برنامج المساعد (Assistant) فى الكتاب الاول .

٣٠ - الأمر (CREATE/MODIFY SCREEN)

يستخدم هذا الأمر فى انشاء شاشات الادخال الجديدة أو تعديل شاشات سبق انشاؤها . وهو يؤدى الى انشاء ملفين أحدهما بالامتداد (.scr) والآخر بالامتداد (.fmt) . حيث يساعد الملف ذو الامتداد (.scr) علي تعديل شاشة الادخال ثم تخزينها فى الملف ذى الامتداد (.fmt) .

والصورة العامة للأمر كالآتى :

CREATE/MODIFY SCREEN <filename>/?

ويستخدم الأمر (CREATE) فى انشاء ملف تشكيل (Format File) جديد . كما يستخدم الأمر (MODIFY) فى انشاء أو تعديل ملف تشكيل سبق ادخاله . ويكتب اسم الملف بدون الامتداد ، حيث أن البرنامج يضيف اليه الامتداد (.scr) . واذا لم يتذكر المستخدم اسم ملف التشكيل المطلوب تعديله يمكنه كتابة (?) بدلا من اسم الملف لعرض جميع ملفات التشكيل المخزنة على القرص أو فى الكتالوج المفتوح ، حيث يتم اختيار الملف المطلوب تعديله . وعند استخدام هذا الأمر تظهر الشاشة التى تظهر عند الدخول فى قائمة الانشاء واختيار (Format For Screen) .

تحذير

فى حالة مسح الملف ذى الامتداد (.scr) لايمكن تعديل ملف التشكيل ذى الامتداد (.fmt) المقابل له . حيث أن ملف التشكيل ذى الامتداد (.fmt) لايمكن ترجمته الى ملف الشاشة ذى الامتداد (.scr) . ولكن فى هذه الحالة يمكن تعديله كملف برنامج باستخدام الأمر (MODIFY COMMAND) . وعند تعديل ملف التشكيل ذى الامتداد (.fmt) بواسطة الأمر (MODIFY COMMAND) ، فلا يمكن تعديله بعد ذلك باستخدام الأمر (MODIFY SCREEN) .

ولمزيد من المعلومات عن هذا الأمر ارجع الى الجزء الخاص بانشاء شاشة ادخال البيانات من خلال برنامج المساعد (Assistant) فى الكتاب الاول .

٣١ - الأمر (CREATE/MODIFY VIEW)

يستخدم هذا الأمر فى ربط عدة ملفات قواعد بيانات عن طريق ملف واحد يسمى ملف المنظر (View File) . وهذا الملف يحتوى على حقول يتم اختيارها من عدة ملفات . ويستخدم هذا الملف الجديد فى عرض أى بيانات من هذه الملفات أو تعديلها .

والصورة العامة للأمر كالآتى :

CREATE/MODIFY VIEW <filename>/?

ويستخدم الأمر (CREATE) فى انشاء ملف المنظر (View File) كما يستخدم الأمر (MODIFY) فى انشاء أو تعديل أى ملف منظر سبق انشاؤه . ويتم كتابة اسم الملف بدون الامتداد حيث أن البرنامج يضيف الامتداد (.VUE) آليا . واذا لم يتذكر المستخدم اسم الملف المطلوب تعديله يمكنه كتابة الحرف (?) مكان اسم الملف فتظهر قائمة بأسماء ملفات المنظر الموجودة على القرص أو فى الكتالوج المفتوح .

وعند تخزين ملف المنظر الذى تم انشاؤه يصبح الملف مفتوحا . كما يمكن فتح أى ملف منظر بكتابة الأمر (SET VIEW TO <filename>) . كما يمكن اغلاق أى ملف منظر مفتوح بكتابة الأمر (CLOSE DATABASES) وهذا يؤدى الى اغلاق جميع ملفات قواعد البيانات والملفات المرتبطة بها مثل ملفات الفهرس والتشكيل.. الخ . كما يؤدى الى اغلاق ملف المنظر الذى يربط هذه الملفات .

وعند كتابة هذا الأمر يظهر عمود الاختيارات (Menu Bar) الذى يحتوى على الاختيارات التالية :

١ - الاختيار (Set Up)

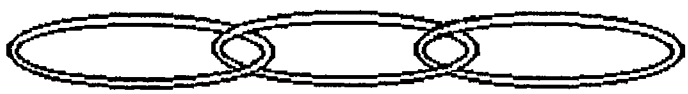
وهذا الاختيار يعرض على الشاشة قائمة بأسماء ملفات قواعد البيانات الموجودة على القرص أو فى الكتالوج المفتوح حتى يستطيع المستخدم اختيار الملفات المطلوب ادخالها فى ملف المنظر . ولاختيار أى ملف من الملفات يتم تحريك العمود الضوئى (Highlight) الى اسم هذا الملف ثم الضغط على

مفتاح الادخال . وهذا يؤدى الى ظهور مثلث يسار اسم الملف . كما يمكن ازالة هذا المثلث بالضغط على مفتاح الادخال مرة ثانية . وهذا يعنى عدم اختيار هذا الملف . وكل ملف يتم اختياره يتم اختيار ملف الفهرس الخاص به . ويمكن ادخال حتى تسعة ملفات قواعد بيانات والملفات الملحقة بها فى ملف منظر واحد .

٢ - الاختيار (RELATE)

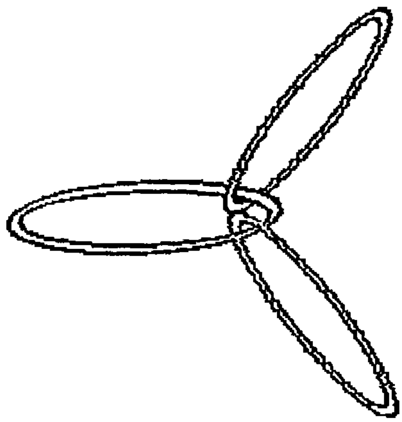
ويستخدم فى تحديد الحقول التى يتم عن طريقها ربط الملفات ببعضها . ويتم عن طريق هذه القائمة عرض الملفات التى سبق اختيارها حتى يتم فتح الملفات التى سوف تستخدم فى تكوين سلسلة العلاقة (Relation Chain) . وترتيب فتح هذه الملفات مهم جدا . حيث أن أول ملف يتم فتحه يعتبر بداية السلسة (Chain).

وبعد فتح هذا الملف تظهر قائمة بباقى الملفات التى سبق اختيارها من خلال قائمة التجهيز (Set Up) فيتم تحديد الملف المطلوب ربطه بالملف الأول . ثم يسأل البرنامج عن اسم الحقل الذى يتم الربط بناء عليه فيتم اختيار اسم هذا الحقل . وبنفس الطريقة يمكن ربط باقى الملفات عندما يراد ربط أكثر من ملفين فى نفس السلسلة (Chain) . ويجب ملاحظة أن الربط يتم على التوالى . فلا يجوز مثلا ربط ملف جديد بملف سبق ربطه بملفات تالية . أى أن السلسلة يجب أن تكون على الشكل التالى:



شكل (٢٧ - ١٧)

وليست كالشكل التالى :



شكل (٢٧ - ١٨)

ويمكن ربط اللفين من خلال مشيرة النقطة (Dot Prompt) أو من خلال البرنامج عن طريق كتابة السطور التالية :

```
SELECT 1
USE file1
SELECT 2
USE file2
SELECT 1
SET RELATION TO field1 INTO file2
```

حيث (field1) هو الحقل المشترك بين اللفين (file1) ، (file2).

٣ - الاختيار (Set Fields)

وهذا الاختيار يسمح للمستخدم باختيار أسماء الحقول المطلوب ادخالها فى ملف المنظر . وفى هذه الحالة يتم اختيار كل ملف من الملفات المرتبطة فى العلاقة فتظهر أسماء الحقول الخاصة بهذا الملف . ويتم اختيار جميع الحقول المطلوبة . ويلاحظ أن الحقول تظهر فى البداية وأمامها علامة (▶) دليل على اختيارها . ولالغاء اختيار أى حقل يتم الضغط على مفتاح الادخال عند وقوف

المؤشر على اسم هذا الحقل .

٤ - الاختيار : (Options)

وهى قائمة تحتوى على الاختيار (Format) الذى يستخدم فى تعديل ملف التشكيل (Format File) الذى سبق انشاؤه لملف المنظر . واذا أريد انشاء ملف تشكيل جديد يتم انشاؤه من خلال قائمة الانشاء (Create) كما سبق الايضاح .

وتحتوى هذه القائمة أيضا على الاختيار (Filter) . وهذا الاختيار يتيح للمستخدم ادخال الشروط المطلوب ادخالها للبحث عن السجلات .

٥ - الاختيار (Exit)

وهو يتيح للمستخدم تخزين ملف المنظر الذى تم انشاؤه من خلال الاختيار (Save) أو الخروج دون تخزينه (Abandon) .

٢٢ - الأمر (CREATE VIEW FROM ENVIRONMENT)

يستخدم هذا الأمر فى انشاء ملف منظر (View File) من العلاقات (Relations) التى سبق انشاؤها من خلال أوامر البرنامج . ويفيد هذا الأمر فى ربط الملفات وانشاء ملف منظر لها من خلال البرنامج دون الرجوع الى برنامج المساعد

(Assistant) والصورة العامة لهذا الأمر كالآتى :

```
CREATE VIEW <filename> FROM ENVIRONMENT
```

وعند كتابة هذا الأمر يتم تخزين جميع ملفات قواعد البيانات وملفات الفهرس والعلاقات التى تم انشاؤها بينها عن طريق الأمر (SET RELATION TO) ، كمايتم ادخال جميع الحقول الخاصة بهذه الملفات فى ملف المنظر ، وذلك فى حالة عدم كتابة الأمر (SET FIELDS TO) الذى يمكن من خلاله تحديد حقول معينة .

٢٣ - الأمر (DELETE)

يستخدم هذا الأمر في تحديد السجلات المطلوب مسحها . وهو يؤدى نفس العمل الذى يتم تنفيذه فى حالة استخدام برنامج المساعد (Assistant) والدخول فى قائمة التحديث (Update) واختيار (Delete) . والصورة العامة للأمر كالآتى :

DELETE [<scope>] [WHILE <condition>] [FOR <condition>]

ويلاحظ هنا أن جميع الاختيارات اختيارية . أى يمكن كتابة الأمر دون كتابة أى شىء بعده . وفى هذه الحالة يتم مسح السجل الذى يقف عنده مؤشر السجلات (Record Pointer) فى هذا الوقت . ويجب ملاحظة أن هذا الأمر لا يمسح السجلات مباشرة ولكنه يضع علامة عندها حتى يتم مسحها نهائيا باستخدام الأمر (PACK) . ويلاحظ عند عرض السجلات باستخدام الأمر (DISPLAY) والأمر (LIST) ، ظهور علامة (*) أمام كل سجل سبق مسحه باستخدام الأمر (DELETE) وهذا يعنى أن السجل موجود ولكنه جاهز للمسح .

والاختيار (scope) يستخدم لتحديد المدى الذى يتم البحث خلاله عن السجلات المطلوب مسحها . كما يستخدم الاختيار (WHILE <condition>) لتحديد الشرط أو الشروط التى يتم البحث بناء عليها . وكذلك يستخدم الاختيار (FOR <condition>) لنفس الغرض .

مثال :

لتجهيز أول سجل فى قاعدة البيانات (Cadets) للمسح يتم كتابة الأوامر التالية :

```
. USE Cadets
. DELETE
1 record deleted
```

يلاحظ ظهور الرسالة الدالة على تجهيز سجل واحد للمسح . كما يمكن كتابة السطر التالى لتجهيز السجل رقم ٨ للمسح .

```
. DELETE RECORD 8
  1 record deleted
```

ويجب ملاحظة أن (RECORD 8) هنا تمثل المدى (Scope) .

كما يمكن كتابة السطر التالى لمسح جميع السجلات التى يبدأ حقل الاسم فيها بالاسم (Mohamed) مثلا :

```
DELETE ALL FOR name = 'Mohamed'
```

ويجب ملاحظة أن (ALL) هنا تمثل المدى (scope)، حيث يتم البحث خلال ملف قاعدة البيانات كله .

٣٤ - الأمر (DIR)

يستخدم هذا الأمر فى عرض دليل ملفات قواعد البيانات الموجودة على وحدة الأقراص المستخدمة أو الفهرس الفرعى المستخدم . والصورة العامة له كالآتى :

```
DIR [<drive:>] [<path>]
```

وكتابة الأمر دون كتابة أي شيء بعده تؤدى الى ظهور أسماء ملفات قواعد البيانات الموجودة على وحدة الأقراص الحالية أو الفهرس الفرعى الحالى . كما يمكن استخدام هذا الأمر فى عرض جميع الملفات عن طريق اضافة الحروف الشاملة (Global Characters) وذلك كالآتى مثلا :

```
DIR *.*
```

وهذا يؤدى الى عرض جميع الملفات الموجودة فى وحدة الأقراص أو الفهرس الحالى سواء كانت ملفات قواعد بيانات أو أى ملفات أخرى .

كما يمكن عرض ملفات الفهرس فقط عن طريق كتابة السطر التالى :

```
DIR *.NDX
```

كما يمكن استخدام المسار (Path) كالآتى مثلا :

```
DIR SALES \*.*
```

ويؤدى هذا الى عرض جميع الملفات الموجودة فى الفهرس الفرعى (SALES) .

كما يمكن عرض ملفات قواعد البيانات الموجودة فى هذا المسار كالآتى :

```
DIR SALES\
```

٣٥ - الأمـــر (DISPLAY)

ويستخدم هذا الأمر فى عرض بيانات الســجلات المطلوبة . والصورة العامة له كالآتى :

```
DISPLAY [<scope>][<expression list>]
[WHILE <condition>][FOR <condition>][ OFF][ TO PRINT]
```

ويلاحظ هنا أن جميع الاختيارات اختيارية حيث يمكن كتابة الأمر دون كتابة أى شيء بعده . وفى هذه الحالة تظهر بيانات السجل الذى يقف عنده مؤشر السجلات (Record Pointer) . كما يتم عرض جميع الحقول الخاصة بهذا السجل .

وعنـد زيادة بيانات الحقول عن عرض الشاشة تنتقل البيانات الى السطر التالى . وعند عرض عدد من السجلات يزيد عن طول الشاشة ، فان عرض السجلات يتوقف مؤقتا (Pause) حتى يضغط المستخدم على أى مفتاح لاستكمال العرض .

والاختيار <scope> يستخدم فى تحديد المدى المطلوب البحث خلاله .

والاختيار <expression list> يستخدم فى تحديد الحقول المطلوب عرض بياناتها .

والاختيار (WHILE <condition>) يستخدم فى تحديد السجلات المطلوب عرض بيانــاتها والتى تحــقق الشرط أو الشروط المكتوبة . وكــذلك الاختـيار (FOR <condition>) يؤدى نفس الغرض . ولكن الاختيار (WHILE) أقوى وأسرع فى الوصول الى السجلات المطلوبة .

والاختيار (OFF) يستخدم عندما يراد عدم ظهور أرقام السجلات على الشاشة .

والاختيار (TO PRINT) يستخدم عندما يراد طباعة البيانات المعروضة ويجب ملاحظة أن أسماء الحقول تظهر فوق البيانات المعروضة . وعندما يراد عدم عرض أسماء الحقول يستخدم الأمر (SET HEADING OFF) . كما أن اسماء الحقول تظهر بالحالة التى يتم كتابتها بها فى الأمر ، أى اذا تمت كتابتها بحروف كبيرة (Capital) تظهر على الشاشة بحروف كبيرة أيضا .

فمثلا لعرض حقول الاسم والعنوان فى ملف الطلبة (Cadets) . يتم كتابة السطور التالية :

```
USE CADETS
DISPLAY NEXT 3 NAME, ADDRESS
```

وفى هذه الحالة تظهر الشاشة التالية :

```
Record #   NAME        ADDRESS

1          MOHAMED     12 - Ahram Street
2          TAREK       8  - Tayaran Street
3          GALAL       20 - Gomhorya Square
```

ويلاحظ هنا أن أسماء الحقول في رأس القائمة (Heading) مكتوبة بحروف كبيرة لأنها تم كتابتها فى الأمر بحروف كبيرة (Capital) .

ملاحظة

حقول الملاحظات (memo fields) لاتظهر محتوياتها على الشاشة الا عند كتابتها منفصلة في الأمر . وفى هذه الحالة تظهر محتويات هذا الحقل فقط على الشاشة . فمثلا اذا كان هناك حقل ملاحظات يسمى (Notes) ، فلكي يتم عرض البيانات المخزنة فيه بالنسبة للسجل الذى يقف عنده مؤشر السجلات (Record Pointer) يتم كتابة الأمر التالى :

```
DISPLAY Notes
```

وفى هذه الحالة تظهر الملاحظات الخاصة بهذا السجل بعرض (٥٠) حرفا فى السطر . واذا اريد تغيير هذا العرض يستخدم الأمر (SET MEMOWIDTH) فى تحديد طول السطر فى حقل الملاحظات .

٣٦ - الأمر (DISPLAY HISTORY)

يستخدم هذا الأمر فى عرض آخر عشرين أمرا تم تنفيذها . والصورة العامة له كالآتى :

DISPLAY HISTORY [LAST <N>] [TO PRINT]

والاختيار (LAST<N>) يستخدم لتحديد عدد الأوامر المطلوب عرضها على الشاشة . وهذا العدد لا يزيد عن عشرين أمرا حيث أنه هو العدد المبدئى (Default). كما يمكن زيادة هذا العدد عن طريق الأمر (SET HISTORY TO) .

والاختيار (TO PRINT) يستخدم عندما يراد طباعة هذه الأوامر على الطابعة .

٣٧ - الأمر (DISPLAY MEMORY)

يستخدم هذا الأمر في عرض حالة متغيرات الذاكرة (Memory Variables) الموجودة فى الذاكرة المؤقتة (RAM) فى أى وقت . حيث يعرض اسم كل متغير ونوعه وحجمه . كما يعرض أيضا بيانات عن حجم الذاكرة المستخدمة وحجم الذاكرة المتاحة . والصورة العامة لهذا الأمر كالآتى :

DISPLAY MEMORY [TO PRINT]

ويلاحظ هنا وجود اختيار واحد فقط وهو (TO PRINT) . وهو يستخدم عندمايراد طباعة هذه البيانات على الطابعة . ويمكن استخدام حتى (٢٥٦) متغير ذاكرة بحد أقصى (٦٠٠٠) حرف . كما يمكن زيادة هذا العدد من خلال ملف المواصفات (Config.sys) . وعندما تزيد البيانات المعروضة عن طول الشاشة يتوقف العرض (Pause) وتظهر الرسالة "Press any key to continue" .

٣٨ - الأمر (DISPLAY STATUS)

يستخدم هذا الأمر فى عرض بيانات عن الحالة الحالية للبرنامج . والصورة العامة له كالآتى :

DISPLAY STATUS [TO PRINT]

وهذا يؤدى الى عرض بيانات عن ملفات قاعدة البيانات المفتوحة ورقم منطقة العمل (Work area) المستخدمة والعلاقات المستخدمة بين الملفات ان وجدت وملفات الفهرس المفتوحة و ... الخ .

٣٩ - الأمر (DISPLAY STRUCTURE)

يستخدم هذا الأمر في عرض بيانات عن ملف قاعدة البيانات المفتوح وتركيبه (Structure) الذى يشمل اسم كل حقل ونوعه وعرضه . كما يعرض عدد الحروف (Bytes) التى يتكون منها السجل الواحد . والصورة العامة لهذا الأمر كالآتى :

```
DISPLAY STRUCTURE  [TO PRINT]
```

وينتج العدد الكلي للحروف (Bytes) في السجل الواحد من مجموع عدد الحروف في جميع الحقول المكونة لهذا السجل بالاضافة الى حرف آخر يتم حجزه لعلامة المسح التى توضع أمام السجل عندما يراد تجهيزه للمسح .

وعند زيادة عدد سطور السجل عن طول الشاشة يتوقف عرض السطور وتظهر الرسالة " Press any key to continue" . ثم يتم استكمال عرض بيانات السجل عند ضغط المستخدم على أى مفتاح .

٤٠ - الأمـر (DO)

يستخدم هذا الأمر فى تنفيذ برنامج فرعى (Module) أو برنامج خطوات (Procedure) . كما يسمح بادخال المعاملات (Parameters) المطلوبة فى البرنامج . والصورة العامة للأمر كالآتى :

```
DO <filename>   [WITH <parameter list>]
```

ويجب أن يتضمن اسم الملف (filename) رمز وحدة الأقراص أو الفهرس الفرعى الذى يحتوى على ملف البرنامج المطلوب . ولايتم كتابة الامتداد فى اسم الملف . حيث أن البرنامج يضيف الامتداد (.prg) آليا .

والاختيار (WITH <parameter list>) يستخدم لادخال معاملات (Parameters) الى البرنامج الفرعى الذى يتم تنفيذه . وعندما ينتهى تنفيذ البرنامج الذى تم استدعاؤه بواسطة الأمر (DO) ينتقل التحكم الى البرنامج الذى قام باستدعائه .

ملاحـــظة

عدد البرامج التى يتم استدعاؤها بواسطة الأمر (DO) يحسب ضمن عدد الملفات المسموح بفتحها فى نفس الوقت . ولكن فى حالة استخدام ملف الخطوات أو الاجراءات (Procedure File) فان فتح هذا الملف يحسب كملف واحد مفتوح بالرغم من أنه يحتوى على العديد من البرامج التى يمكن تنفيذها بواسطة الأمر (DO) لذلك يفضل استخدام ملف الخطوات أو الاجراءات (Procedure File) على استخدام البرامج المنفصلة (Modules) فى أغلب الأحيان .

مـــثال ١

لتنفيذ البرنامج (Cadrep) من داخل البرنامج (Cadets) مثلا يتم كتابة الأمر التالى :

```
DO B: Cadrep
```

مثـــال ٢

لكتابة البرنامج (Calc) الذى يحسب مساحة المستطيل ، فان هذا البرنامج يكون كالآتى مثلا :

```
PARAMETERS Length, Width, Area
Area = Length * Width
RETURN
```

ولكى يتم تنفيذ هذا البرنامج مع ادخال الأطوال 8 ، 5 مكان المعاملات (Length) ، (Width) على الترتيب ، يتم كتابة السطور التالية :

```
Area = 0
DO Calc WITH 8,5 , Area
? Area
```

وعند الضغط على مفتاح الادخال يظهر العدد (40).

والسطر الأول يتم كتابته لحجز مكان فى الذاكرة للمتغير (Area) .

٤١ - الأمر (DO CASE)

يستخدم هذا الأمر فى الانتقال بين مجموعات مختلفة من الأوامر بناء على شروط معينة يتم ادخالها . وهو يعتبر صورة مكبرة من الأمر (IF) ، حيث أن الأمر (IF) يسمح بالاختيار بين بديلين فقط ، أما هذا الأمر فيسمح بالاختيار بين عدة بدائل . والصورة العامة له كالآتى :

```
DO CASE
   CASE <condition>
        <commands>
   CASE<condition>
        <commands>
[OTHERWISE]
        <commands>
ENDCASE
```

ويلاحظ أنه يتم التفرع الى أى حالة (CASE) عندما يصبح الشرط (condition) الخاص بها صحيحا (True) . والشرط (condition) هو عبارة عن علاقة منطقية (Logical) مثل (A = B) أو (mchoice = 1) . ويبدأ البرنامج باختبار الشرط الأول . فاذا تحقق هذا الشرط يتم تنفيذ الأوامر (Commands) التالية له . واذا لم يتحقق ينتقل البرنامج الى الشرط الثانى ... وهكذا .

وعندما ينفذ البرنامج مجموعة من الأوامر التابعة لحالة معينة (CASE) ، فانه لايقوم باختبار باقى الحالات . وانما ينتقل الى الأوامر التالية للأمر (ENDCASE) .

والاختيار (OTHERWISE) اختيارى يمكن عدم كتابته . ويستخدم عندما يراد تنفيذ مجموعة من الأوامر فى حالة عدم تحقق أى شرط من الشروط السابقة .

٤٢ - الأمر (DO WHILE)

يستخدم هذا الأمر فى تكوين حلقة تكرارية (Loop) يتم تنفيذها أى عدد من المرات طالما كان الشرط الموجود بعد الأمر صحيحا (True) . والصورة العامة لهذا الأمر كالآتى :

```
DO WHILE <condition>
    <commands>
ENDDO
```

ويستخدم الأمر (ENDDO) لتحديد نهاية الحلقة . حيث يعود البرنامج الى أول الحلقة ليختبر الشرط مرة ثانية . فاذا تحقق الشرط يتم تنفيذ أوامر الحلقة . واذا لم يتحقق يتم الانتقال الى الأوامر التى تلى الأمر (ENDDO) . ويمكن استخدام حلقات متداخلة (Nested) . وفى هذه الحالة يجب التأكد من ملاحظة انتهاء كل حلقة داخل الحلقة الخارجية . ولتنظيم هذه العملية يمكن كتابة ملاحظات (Comments) بجوار كل أمر (ENDDO) لتوضيح الحلقة التى يتبعها مع ملاحظة أن أى تعليق (Comment) بجوار الأمر (ENDDO) يتم اهماله بواسطة البرنامج أى أنه يفيد فى التوضيح فقط فى حالة عرض أوامر البرنامج .

ويمكن استخدام التعويض أو الاحلال بالماكرو (Macro Substitution) ولكن يراعى فى هذه الحالة عدم تغيير قيمة هذا الماكرو داخل الحلقة التكرارية .

والبرنامج التالى مثلا يوضح هذه الطريقة .

```
USE Cadets INDEX Name
STORE (UPPER(name) = 'TAREK') TO condition
DO WHILE & condition .AND. .NOT. EOF()
    ?  TRIM(name), address
    SKIP
ENDDO
USE
RETURN
```

وهذا البرنامج يقوم أولا بتخزين الشرط الموضح فى متغير ذاكرة اسمه (condition) . ثم يقوم باستخدام هذا المتغير فى الشرط الموجود بعد الأمر .(DO WHILE)

٤٣ - الأمر (EDIT)

يستخدم هذا الأمر فى تعديل بيانات سجل معين فى ملف قاعدة البيانات . وهو يؤدى نفس العمل الذى يتم عند استخدام برنامج المساعد (Assistant) عن طريق الدخول الى قائمة التحديث (Update) واختيار (EDIT) .

والصورة العامة لهذا الأمر كالآتى :

```
EDIT[<scope>][FIELDS <list>][WHILE <condition>]
    [FOR <condition]
```

ويلاحظ أن جميع الاختيارات اختيارية أى يمكن عدم استخدامها . وفى حالة كتابة الأمر (EDIT) دون كتابة أى شيء بعده تظهر البيانات الخاصة بالسجل الذى يقف عنده مؤشر السجلات (Record Pointer) . وفى هذه الحالة يمكن تعديل البيانات عن طريق تحريك مؤشر التصحيح الى الحقل المطلوب تعديله وكتابة البيانات الجديدة . ويتم الانتقال من سجل الى آخر باستخدام مفتاحى (PgUp) ، (PgDn) . كما يتم تخزين التعديلات التى تم ادخالها بالضغط على مفتاحى (Ctrl-End) .

ولتعديل حقل الملاحظات (memo field) يتم وضع مؤشر التصحيح على هذا الحقل . ثم بالضغط على مفتاحى (Ctrl-PgDn) يتم فتح حقل الملاحظات حيث يتم تعديل الملاحظات أو اضافة ملاحظات جديدة . كما يتم تخزين هذه الملاحظات بالضغط على مفتاحى (Ctrl-PgUp) أو مفتاحى (Ctrl-End) .

٤٤ - الأمر (EJECT)

يستخدم هذا الأمر فى ارسال شفرة نقل الصفحة (Form Feed) الى الطابعة ، وهى عبارة عن شفرة الآسكى (ASCII 12) . ويراعى فى بداية الطباعة أن تبدأ الورقة من أولها .

والصورة العامة للأمر كالآتى :

```
EJECT
```

وهذا الأمر يعيد عداد السطور (Prow()) وعداد الأعمده (Pcol()) الى الصفر .

٤٥ - الأمر (ERASE)

يستخدم هذا الأمر فى مسح ملف من القرص أو من الفهرس الفرعى . والصورة العامة له كالآتى :

```
ERASE <filename>/?
```

واسم الملف يجب أن يتضمن الامتداد . كما يجب كتابة وحدة الأقراص أو الفهرس الفرعي في حالة عدم وجود الملف المطلوب مسحه على وحدة الأقراص الحالية (Current Drive) . وهذا الأمر يقوم بتنفيذ نفس العمل الذى يتم عند استخدام برنامج المساعد (Assistant) والدخول فى قائمة الأدوات (Tools) ، ثم اختيار (ERASE) . كما أن استخدام هذا الأمر يماثل تماما استخدام الأمر (DELETE FILE) .

ويجب ملاحظة أن هذا الأمر لايسمح باستخدام الحروف الشاملة لمسح عدد من الملفات مثل أوامر نظام التشغيل (MS-DOS) .

٤٦ - الأمر (EXIT)

يستخدم هذا الأمر فى الخروج من الحلقة التكرارية التى تتكون نتيجة لاستخدام الأمر (DO WHILE) . ويؤدى هذا الأمر الى الانتقال الى الأوامر التى تلى الأمر (ENDDO) .

والصورة العامة له كالآتى :

```
EXIT
```

٤٧ - الأمر (FIND)

يستخدم هذا الأمر للبحث في ملف قاعدة البيانات المفهرس (Indexed) عن أول سجل يماثل الحقل الفهرسى فيه سلسلة حرفية أو عدد معين يتم ادخاله . وهو يوفر سرعة بحث عالية جدا .

والصورة العامة له كالآتى :

```
FIND <character string> /<n>
```

واذا لم تكن السلسلة الحرفية أو العدد الذي يجري البحث عنه موجودا في قاعدة البيانات تظهر العبارة "No found" على الشاشة ، وفى هذه الحالة يصبح مؤشرالسجلات (Record Pointer) عند آخر الملف ، أى أن الداله (EOF()) تصبح صحيحة (True) . وعادة لاتوضع السلسلة الحرفية التي يتم البحث عنها بين علامات تنصيص (Quotation Marks) . ولكن اذا كانت السلسلة الحرفية تحتوى على مسافات خالية فى أولها ، فيجب فى هذه الحالة وضعها بين علامات تنصيص . ويجب ملاحظة أن المقارنة تصبح صحيحة اذا كانت الحروف الأولى فى

السلسلة الحرفية المطلوبة مطابقة للحروف الأولى من حقل المفتاح (Index Key) ، بصرف النظر عن باقي الحروف .

وعند البحث عن متغير ذاكرة حرفي يجب اضافة الدالة (&) قبل اسم المتغير . واذا كان هذا المتغير يحتوى على مسافات في بدايته يتم وضع الاسم مع الدالة (&) بين علامات تنصيص (Quotation Marks) ، كالآتى مثلا "&mname" .

والأمر (FIND) يماثل الأمر (SEEK) ، ولكن الأمر (SEEK) أشمل منه . حيث يمكن استخدام العلاقات الحسابية معه كما سبق الايضاح فى الكتاب الاول.

مثال

للبحث عن أى اسم يبدأ بالحرف (M) فى ملف الطلبة (Cadets) يتم كتابة السطور التالية :

```
.  USE Cadets INDEX Name
.  FIND M
.  ? name
Mohamed
```

يلاحظ فى هذه الحالة ظهور الاسم (Mohamed) عند السؤال عن الاسم الموجود فى الحقل الفهرسي (name) باستخدام الأمر (?) . ويلاحظ أن المقارنة قد تمت بين الحرف المطلوب والحرف الأول فى حقل الاسم فقط . ولكن فى حالة استخدام الأمر(SET EXACT ON) ، يصبح الأمر مختلفا . فمثلا عند كتابة الأوامر السابقة بالصورة الآتية :

```
. USE Cadets INDEX Name
. SET EXACT ON
. FIND M
No found
```

يلاحظ فى هذه الحالة ظهور العبارة "No found" . وذلك لعدم وجود الاسم المطابق تماما للحرف (M) ، أى الذى يتكون من الحرف (M) فقط .

٤٨ - الأمر (GO/GOTO)

يستخدم هذا الأمر فى توجيه مؤشر السجلات (Record Pointer) الى سجل معين . وهناك صورتان لهذا الأمر :

الصورة الأولى تكون كالآتى :

```
[GO/GOTO]   <N>
```

حيث (N) هو رقم السجل المراد توجيه المؤشر اليه . ويلاحظ أن الأمر (GO/GOTO) اختيارى فى هذه الحالة . أى يكفى كتابة الرقم فقط دون كتابة الأمر كالآتى مثلا :

```
.    USE Cadets
   8
.  ? RECNO()
   8
```

يلاحظ هنا عند السؤال عن رقم السجل الذى يقف عنده المؤشر باستخدام الدالة (RECNO()) ظهور الرقم (٨) .

كما يمكن استخدام متغيرات الذاكرة مع الأمر (GO/GOTO) . وذلك كالآتى مثلا :

```
.  STORE 5 TO mnum
.  GO mnum
.  ? RECNO()
   5
```

والصورة الثانية تكون كالآتى :

```
GO/GOTO BOTTOM/TOP
```

ويلاحظ هنا أن الأمر (GO/GOTO) ليس اختياريا كالحالة السابقة . أى أن وجوده ضرورى فى هذه الحالة .

مثال

```
.    GO TOP
.    ? RECNO()
     1
```

ويلاحظ ظهور الرقم (1) عند السؤال عن رقم السجل .

٤٩ - الأمــر (HELP)

يستخدم هذا الأمر فى عرض شاشات المساعدة التى يتم من خلالها شرح أوامر البرنامج . والصورة العامة للأمر كالآتى :

```
HELP keyword
```

حيث keyword هو أي أمر أو دالة من برنامج (+DBase III). ويمكن الضغط على مفتاح (F1) الذى يؤدى نفس العمل مثل كتابة الأمر (HELP) . ويؤدى هذا الى ظهور القائمة الرئيسية لشاشات المساعدة التى يمكن من خلالها الوصول الى الأمر المطلوب ومتابعة الشرح الخاص به .

٥٠ - الأمــر (IF)

ويستخدم هذا الأمر فى تنفيذ مجموعة من الأوامر فى حالة تحقق شرط معين . والصورة العامة له كالآتى :

```
IF    <condition>
      <commands>
[ELSE]
      <commands>
ENDIF
```

ويقوم البرنامج باختبار الشرط (condition) بعد (IF) فاذا تحقق يتم تنفيذ الأوامر (Commands) التالية للأمر (ELSE) ، ويلاحظ أن الأمر (ELSE) اختيارى هنا . أى يمكن عدم كتابته ، وفى هذه الحالة يتم الانتقال الى الأوامر التى تلى الأمر (ENDIF) فى حالة عدم تحقق الشرط .

ويمكن استخدام مجموعات متداخلة من الأوامر (IF) . وفى هذه الحالة يجب ملاحظة وجود الأمر (ENDIF) الخاص بكل مجموعة داخل المجموعة الخارجية . ويمكن ملاحظة ذلك عن طريق كتابة تعليق (Comment) بعد الأمر (ENDIF) . حيث أن هذا التعليق لايؤثر فى تنفيذ البرنامج .

٥١ - الأمــر (INDEX)

يستخدم هذا الأمر فى انشاء ملف فهرسي يحتوى على حقلين فقط ، أحدهما يحتوى على أرقام السجلات الموجودة فى ملف قاعدة البيانات ، والآخر يحتوى على

محتويات الحقل الفهرسي (Index Key) الخاصة بكل سجل . والصورة العامة لهذا الأمر كالآتي :

```
INDEX ON <index key> TO <filename>[UNIQUE]
```

حيث :

(Index Key)	هو المفتاح الذى يتم الترتيب بناء عليه ، وهو قد يكون حقلا معينا أو علاقة تشمل عدة حقول . ويجب ملاحظة أن حقول اللاحظات (memo fields) لاتستخدم في هذا المفتاح وكذلك الحقول المنطقية (Logical fields).
و (Filename)	هو اسم الملف الذي يتم انشاؤه وهو يشمل رمز وحدة الأقراص المستخدمة اذا كان الملف سيتم تخزينه في وحدة أقراص غير الوحدة الحالية (current drive). كما أن الاسم يضاف اليه الامتداد (.NDX) آليا .

والاختيار (UNIQUE) هنا اختياري وهو يعنى أنه لاتوجد داخل الفهرس سوى قيمة واحدة لكل سجل . ويعني ذلك أنه اذا كان هناك أكثر من سجل يحتوى على نفس القيمة للحقل الفهرسي يتم ادخال أول سجل يحتوى على هذه القيمة في ملف الفهرس .

ويجب ملاحظة أن الفهرسة تكون دائما تصاعدية (Ascendingly) . كما أنها لاتغير الترتيب الفعلي للسجلات . ولكنها تقوم بترتيب السجلات عند عرضها أو التعامل معها فى الذاكرة المؤقتة . أى أنها تعطى صورة مرتبة للسجلات مع الاحتفاظ بالأصل في الملف دون أى تغيير .

ويمكن اضافة مجموعة من الحقول في الحقل الفهرسى . وعند اضافة أنواع مختلفة من الحقول مثل الحقول العددية والحقول التاريخية ، يتم تحويل كل نوع من هذه الأنواع الى حقول حرفية . حيث يتم تحويل الحقول العددية عن طريق الدالة (STR) ، كما يتم تحويل الحقول التاريخية عن طريق الدالة (DTOC) . واقصى طول للحقل الفهرسى هو (١٠٠) حرف .

فمثلا اذا أريد انشاء ملف فهرسى يحتوى على التاريخ والاسم يتم كتابة السطر التالي :

```
INDEX ON STR(YEAR(<date>),4)+STR(MONTH(<date>),2)+ ;
STR(DAY(<date>),2 ) + Name TO <filename)
```

وفى هذه الحالة يتم ترتيب الملف بناء على التاريخ أولا ، ثم يتم ترتيب الأسماء داخل كل تاريخ .

٥٢ - الأمـــر (INPUT)

يستخدم هذا الأمر فى توجيه المستخدم الى ادخال قيمة معينة وتخزين هذه القيمة فى متغير ذاكرة . والصورة العامة له كالآتى :

```
INPUT   [<message>]   TO <memvar>
```

حيث :

(message) هى رسالة حرفية تظهر للمستخدم لتوجيهه الى ادخال المتغير المطلوب . وهى تكون محصورة بين علامات تنصيص فردية 'Single quotes' أو مزدوجة "double quotes" أو أقواس مربعة (Brackets) . وقد تستخدم متغيرات الذاكرة مكان هذه الرسالة ، وفى هذه الحالة لاتوضع بين علامات تنصيص (Quotes) .

وعند ادخال المستخدم للقيمة المطلوبة يتم تخزين هذه القيمة فى متغير الذاكرة (memvar) . ويتحدد نـــوع هذا المتغير حسب نوع المدخلات التى يدخلها المستخدم . فمثلا عندما يقوم بادخال عدد ، يتم انشاء متغير عددى لتخزين هذا العدد . ويجب ملاحظة أن القيم الحرفية يتم ادخالها بين علامات تنصيص (Quotes).

وهذا الأمر يشبه الأمر (ACCEPT) ولكن الأمر (ACCEPT) يتعامل مع المدخلات الحرفية فقط .

مثـــال

لتوجيه المستخدم الى ادخال اسمه وتخزين هذا الاسم فى متغير الذاكرة (mname) يتم كتابة السطر التالى :

```
INPUT "What is your name" TO mname
```

٥٣ - الأمر (INSERT)

يستخدم هذا الأمر فى اضافة سجل جديد الى ملف قاعدة البيانات عند المكان الذى يقف عنده مؤشر السجلات (Record Pointer) والصورة العامة له كالآتى :

```
INSERT [BLANK][BEFORE]
```

ويلاحظ هنا أنه يمكن استخدام الأمر (INSERT) بدون أى اختيارات أخرى . وفى هذه الحالة تظهر الحقول الخاصة بسجل جديد يتم ادخال بياناته . وعند الانتهاء من ادخال البيانات يتم تخزين هذه البيانات بالضغط على مفتاحى (Ctrl-End) . وفى هذه الحالة يتم تخزين هذا السجل بعد السجل الحالى (Current Record) الذى يقف عنده المؤشر . وعند استخدام هذا الأمر مع الملف المفهرس (Indexed File) فانه يعمل مثل الأمر (APPEND) تماما . حيث يضيف سجلا فى آخر الملف . ثم يضيف سجلات أخرى واحدا تلو الآخر .

والاختيار (BEFORE) يفتح سجلا قبل السجل الحالى . فمثلا اذا كان السجل الحالى هو السجل رقم ٨ فان الأمر (Insert Before) يفتح سجلا جديدا يكون رقمه (٨) فى حين يصبح السجل السابق رقم (٩) .

والاختيار (BLANK) يؤدى الى اضافة سجل جديد خال مكان السجل الحالى ، بحيث يمكن اضافة البيانات اليه فيما بعد .

٥٤ - الأمر (JOIN)

يستخدم هذا الأمر فى انشاء ملف جديد عن طريق دمج حقول وسجلات من ملفين مفتوحين . وهو يؤدى عمل الأمر (UPDATE) ولكنه يختلف عنه فى أنه ينشىء ملفا جديدا . أما الأمر (UPDATE) فانه يعدل ملفا بناء على ملف آخر .

والصورة العامة لهذا الأمر كالآتى :

```
JOIN WITH <alias> TO <filename>
  FOR <condition>[FIELDS <fieldlist>]
```

حيث (filename) هو الملف الجديد الذى يتم انشاؤه ويجب أن يتضمن رمز وحدة الأقراص اذا كان يراد تخزينه على وحدة أقراص غير وحدة الأقراص الحالية (Current Drive).

و (alias) هو المرادف الذي يشمل اسم الملف المراد دمجه ومنطقة العمل الخاصة بهذا الملف . ويراعى أن يكون الملف المراد دمجه مفتوحا وأن تكون منطقة العمل (Work Area) الخاصة به قد تم اختيارها (Selected) كمنطقة العمل العاملة (Active) .

والاختيار (FIELDS <field list>) يستخدم لتحديد أسماء الحقول المراد دمجها من الملفين . وفى حالة عدم تحديد هذه الحقول يتم دمج جميع الحقول فى الملف الجديد .

ملاحظة

عندما يراد تحديد حقول من منطقة عمل غير مختارة (Unselected Work Area) يستخدم التعبير (Alias --> field name) حيث (Alias) هو الاسم المرادف لمنطقة العمل غير المختارة .

٥٥ - الأمر (LABEL)

يؤدى هذا الأمر الى استخدام ملف العناوين المختصرة (Label) الذى سبق انشاؤه . والصورة العامة له كالآتى :

```
LABEL FORM <filename>/?[<scope>][SAMPLE]
   [WHILE <condition>][FOR <condition>][TO PRINT]
   [TO FILE <flename>]
```

حيث (filename) هو اسم ملف العناوين المختصرة المراد استخدامه ويجب أن يشمل رمز وحدة الأقراص اذا كانت غير وحدة الأقراص المستخدمة (Current drive).

وتستخدم علامة الاستفهام (?) عندما يراد عرض أسماء ملفات العناوين الموجودة على القرص حتى يستطيع المستخدم اختيار الملف المطلوب استخدامه .
و (Scope) هو المدى المطلوب البحث خلاله عن السجلات المطلوب عرض بياناتها فى التقرير .

والاختيار (SAMPLE) يستخدم لاختبار ضبط التقرير على الطابعة . ويمكن تكرار عملية الاختبار عدة مرات بكتابة (Y) عند ظهور السؤال التالى :

```
Do you want more samples?
```

والاختيار (TO FILE) يستخدم عندما يراد تخزين التقرير فى ملف نصوص (Text File) حتي يمكن التعامل معه بواسطة برامج أخرى .

والاختيارات (WHILE) ، (FOR) تستخدم لتحديد السجلات المطلوب عرض بياناتها فى التقرير .

٥٦ - الأمر (LIST)

يستخدم هذا الأمر فى عرض بيانات ملف قاعدة البيانات . والصورة العامة له كالآتى :

```
LIST [OFF][<scope>][<Field list>][WHILE <condition>]
     [FOR <condition>][TO PRINT]
```

ويلاحظ هنا أن جميع الاختيارات مع الأمر اختيارية . أى يمكن كتابته دون كتابة أى شيء بعده . وفي هذه الحالة يتم عرض بيانات جميع سجلات قاعدة البيانات وهو في هذا يختلف عن الأمر (DISPLAY) الذي يعرض بيانات السجل الحالي فقط . كما أن الأمر (LIST) لايوقف الشاشة عند امتلائها بالبيانات ، بل يتم الانتقال الى الشاشات التالية فورا . وعندما يراد ايقاف الشاشة (Pause) يتم الضغط على مفتاحى (Ctrl-S) .

والاختيار (OFF) يستخدم عندما يراد عرض البيانات بدون أرقام السجلات .

والاختيار (scope) يستخدم لتحديد المدى الذى يتم البحث خلاله عن السجلات المراد عرضها .

والاختيار (Field List) يستخدم لتحديد أسماء الحقول المراد عرض بياناتها .

والاختيارات (WHILE) ، (FOR) تستخدم لتحديد السجلات المراد عرض بياناتها .

والاختيار (TO PRINT) يستخدم عندما يراد طباعة البيانات .

٥٧ - الأمــر (LIST HISTORY)

يستخدم هذا الأمر فى عرض آخر عشرين أمرا سبق تنفيذها وهو يماثـل الأمـر (DISPLAY HISTORY) ولكنه لايؤدى الى توقف الشاشة عندما تزيد الأوامـر عـن طول الشاشة . والصورة العامة له كالآتى :

LIST HISTORY [LAST <N>] [TO PRINT]

والاختيار (LAST <N>) يستخدم لتحديد عدد الأوامر المطلوب عرضها .

والاختيار (TO PRINT) يستخدم لطباعة الأوامر المعروضة .

٥٨ - الأمــر (LIST MEMORY)

يستخدم هذا الأمر في عرض حالة متغيرات الـذاكـرة (Memory Variables) الموجودة في الذاكرة المؤقتة (RAM) فى أى وقت ، حيث يعرض اسم كل متغير ونوعه وحجمه ، كما يعرض أيضا بيانات عن حجــم الذاكرة المستخدمة وحجم الذاكرة المتاحة . والصورة العامة لهذا الأمر كالآتى :

LIST MEMORY [TO PRINT]

وهو يماثـل الأمـر (DISPLAY MEMORY) تمـامـا ويختلف عنه فقـط فـى أنـه لايوقف عرض البيانات عندما تزيد عن طول الشاشة .

٥٩ - الأمــر (LIST STATUS)

يستخدم هذا الأمر فى عرض بيانات عن الحالة الحالية للبرنـامج . والـصـورة العامة له كالآتى :

LIST STATUS [TO PRINT]

وهو يؤدى الى عرض بيانات عن ملفات قاعدة البيانات المفتوحة ورقم مـنطقة العمل (Work Area) المستخدمة والعلاقات المستخدمة بين الملفات ان وجـدت وملفـات الفهرس المفتوحة و ... الخ .

وهذا الأمر يماثل الأمر (DISPLAY STATUS) تماما ولكنه لايتوقف عند امتلاء الشاشة بالبيانات .

٦٠ - الأمـر (LIST STRUCTURE)

يستخدم هذا الأمر في عرض بيانات عن ملف قاعدة البيانات المفتوح وتركيبه (Structure) الذى يشمل اسم كل حقل ونوعه وعرضه . كما يعرض عدد الحروف (Bytes) التى يتكون منها السجل الواحد . والصورة العامة لهذا الأمر كالآتى :

```
LIST STRUCTURE  [TO PRINT]
```

وهو يماثل الأمر (DISPLAY STRUCTURE) تماما . ولكنه لايتوقف عند امتلاء الشاشة بالبيانات .

٦١ - الأمر (LOAD)

يستخدم هذا الأمر في كتابة برامج فرعية بلغة التجميع (Assembly) وتشغيل هذه البرامج من خلال البرنامج الذى يتم تصميمه بواسطة (DBase III+) . ويجب أن تكون هذه البرامج على هيئة الشفرة الثنائية (Binary Code) . والصورة العامة لهذا الأمر كالآتى :

```
LOAD <filename>
```

وهذا يؤدى الى تحميل الملف (Filename) فى الذاكرة المؤقتة حتى يتم استدعاؤه فى البرنامج بواسطة الأمر (CALL) . ويسمح برنامج (DBase III+) بتحميل حتى خمسة برامج من هذا النوع في كل مرة . وكل برنامج بحد أقصى ٣٢ ك بايت بشرط ان تكون بالشفرة الثنائية (Binary) .

٦٢ - الأمـر (LOCATE)

يستخدم هذا الأمر للبحث في ملف قاعدة البيانات عن السجل الذى يحقق شرطا أو شروطا معينة . والصورة العامة له كالآتى :

```
LOCATE [<scope>] [WHILE <condition>] [FOR <condition>]
```

والاختيار (scope) يستخدم لتحديد مدى معين من السجلات للبحث خلاله . فمثلا عند كتابة (NEXT <N>) فان البحث يتم خلال العدد (N) من السجلات التى تلى السجل الحالى (Current Record).

والاختيارات (WHILE) ، (FOR) تستخدم لتحديد السجلات المطلوب البحث عنها . ويجب ملاحظة أن هذا الأمر يؤدى الى توجيه المؤشر الى أول سجل يحقق الشروط . فاذا أريد تحريكه الى باقى السجلات يستخدم الأمر (CONTINUE) .

٦٣ - الأمر (LOOP)

يستخدم هذا الأمر فى الانتقال الى بداية الحلقة التكرارية دون تنفيذ باقى أوامر الحلقة . والصورة العامة له كالآتى :

```
LOOP
```

ويستخدم فى معظم الأحيان فى البرامج الفرعية (Modules) الخاصة بقوائم الاختيارات (Menus) ، فمثلا يمكن كتابة الأوامر التالية فى قاعدة بيانات الطلبة (Cadets) :

```
USE Cadets
     mchoice = " "
     CLEAR
     @ 5,5 SAY "A-Add records"
     @ 7,5 SAY "E-Edit records"
     @ 9,5 SAY "R-Return to dot prompt"
     READ
IF .NOT. mchoice $ "AaEeRr"
     LOOP
ENDIF
DO CASE
     CASE mchoice $ "Aa"
          APPEND
     CASE mchoice $ "Ee"
          EDIT
     CASE mchoice $ "Rr"
          EXIT
     ENDCASE
ENDDO
CLOSE DATABASES
```

ويتم اختبار المتغير (mchoice) بعد الأمر (IF) ، فاذا لم يكن الحرف المخزن به ضمن الحروف (AaEeRr) يتم تنفيذ الأمر التالي وهو (LOOP) . وفي هذه الحالة يتم الانتقال الى أول الحلقة التكرارية مرة ثانية لاختبار الحرف الجديد الذى يدخله المستخدم في المتغير (mchoice) . والدالة ($) تعني هنا البحث عن الحرف الموجود في المتغير (mchoice) خلال السلسلة الحرفية (AaEeRr) . وفي هذه السلسلة تم وضع الحروف المثلة لاختيارات القائمة سواء كانت صغيرة (small) أو كبيرة (Capital) .

٦٤ - الأمر (MODIFY COMMAND)

يستخدم هذا الأمر في انشاء وتعديل ملفات الأوامر (Command files) التى تستخدم في كتابة البرامج الخاصة ببرنامج (DBase III+) . وهذا الأمر يؤدى الى تشغيل برنامج تنسيق النصوص (Text editor) الخاص ببرنامج (DBase III+) والذى يمكن استخدامه في كتابة أى نص (Text) .

والصورة العامة للأمر كالآتى :

```
MODIFY COMMAND <filename>
```

ويمكن أن يحتوى اسم الملف (filename) على رمز وحدة الأقراص المراد تخزين الملف عليها اذا لم تكن هى وحدة الأقراص الحالية (Current Drive) . وهذا الأمر يضيف الامتداد (.prg) الى اسم الملف آليا . أما اذا أريد اضافة أى امتداد آخر، فيجب في هذه الحالة اضافة الامتداد الى اسم الملف المراد انشاؤه .

وعند كتابة هذا الأمر فان البرنامج يبحث عن ملف بالاسم (Filename) فاذا وجده فانه يستدعيه ويعرض الملف لتصحيحه . واذا لم يجده فانه ينشيء ملفا جديدا بهذا الاسم .

وهذا الأمر يسمح بانشاء ملف حتى ٥٠٠٠ حرف (Byte) . أما اذا استخدم برنامج تنسيق نصوص خارجي من خلال ملف المواصفات (Configuration File) فانه يمكن في هذه الحالة زيادة عدد الحروف عن ذلك .

وعند كتابة أمر يزيد عن طول السطر يمكن كتابة الفاصلة المنقوطة (;) فى نهاية السطر ثم استكمال كتابة الأمر في السطر التالى .

وعـــندمـــــا يــــــراد طباعـــــــة محـــــــتويات الملـــــف يستخدم الأمر (TYPE <filename> TO PRINT) من مشيرة النقطة (Dot Prompt) .

٦٥ - الأمـــر (MODIFY LABEL)

ارجع الى (CREATE/MODIFY LABEL)

٦٦ - الأمـــر (MODIFY QUERY)

ارجع الى (CREATE/MODIFY QUERY)

٦٧ - الأمـــر (MODIFY REPORT)

ارجع الى (CREATE/MODIFY REPORT)

٦٨ - الأمـــر (MODIFY SCREEN)

ارجع الى (CREATE/MODIFY SCREEN)

٦٩ - الأمـــر (MODIFY STRUCTURE)

ويستخدم هذا الأمر عندما يراد تعديل تركيب ملف قاعدة البيانات والصورة العامة له كالآتى :

```
MODIFY STRUCTURE <Filename>
```

وهذا الأمر ينشيء دائما ملفا احتياطيا بالامتداد (.bak) لكل ملف يتم تعديله . وعند الانتهاء من ادخال التعديلات ، يقوم باضافة السجلات (Append) الى الملف الأصلى . واذا تم تعديل أى حقول ملاحظات ، يتم انشاء ملف احتياطى لملف الملاحظات بالامتداد (.&&&) .

٧٠ - الأمر (NOTE/*)

يستخدم الأمر (NOTE) أو النجمة (*) لنفس الهدف وهو كتابة ملاحظات لتوضيح وظائف أوامر البرنامج المختلفة . ويتم كتابة هذا الأمر فى بداية السطر المطلوب كتابة ملاحظات فيه . والصورة العامة لهذا الأمر كالآتى :

```
NOTE/* <text>
```

وعندما يراد كتابة عدة سطور ملاحظات متتالية ، يمكن كتابة فاصلة منقوطة(;) فى نهاية كل سطر . ويمكن كتابة الملاحظات فى أى مكان فى السطر باستخدام الحرفين (&&) .

مـثال

هذا المثال يوضح استخدام الأمر (NOTE) فى برنامج .

```
NOTE This is a loop
STORE 1 TO X
DO WHILE X < 100
     STORE X + 1 TO X
ENDDO
```

كما يمكن استخدام الحرفين (&&) كالآتى مثلا :

```
mname = SPACE (35) && initiate a memory variable
```

٧١ - الأمـر (ON)

يستخدم هذا الأمر فى تنفيذ أمر أو برنامج فرعى بناء على تحقيق شرط معين . والصورة العامة له كالآتى :

```
ON ERROR/ESCAPE/KEY <command>
```

وهناك ثلاثة صور للأمر كما هو واضح وهى كالآتى :

ON ERROR <command>

وهـو يؤدى الى تنفيذ أمر معين أو مجموعة من الأوامر فى حالة ظهور خطأ (Error) فى البرنامج .

ON ESCAPE <command> وهـو يؤدى الى تنفيذ أمر معين أو مجموعة من الأوامر فى حالة ضغط المستخدم على مفتاح الهروب (ESC) .

ON KEY <command> وهو يؤدى الى تنفيذ أمر معين أو مجموعة من الأوامر فى حالة ضغط المستخدم على أى مفتاح .

ويظل هذا الأمـر مـؤثرا فى البرنامج حتـى انهائـه بواسطـة الأمـر (ON ERROR/ESCAPE/KEY) ، وذلك دون كتابة أى شيء بعده .

ويـجب ملاحـظة أن الأمـر (ON ESCAPE) لايعـمل مـع الأمر (SET ESCAPE OFF) . كما يجب ملاحظة أن الأمر (ON ESCAPE) له أولوية على الأمر (ON KEY) . وهذا يعنى أنه عند كتابة الأمرين معا فى نفس البرنامج فان الضغط عـلى مفتـاح الهـروب يؤدى الى تنفيذ الأمر أو الأوامر التى تلى الأمر (ON ESCAPE) وليس الأوامر التى تلى الأمر (ON KEY) .

والأمر (ON ERROR) يتعامل فقـط مـع الأخطـاء المتعلقة بقواعد أوامر (DBase III+) ، وهـى التى تسمى أخطاء القواعد (Syntax Errors) .

مـثال

يمكن تشغيل البرنامج (error1) عند ظهور أى خطأ فى البرنامج ، وذلك عن طريق كتابة السطر التالى :

```
ON ERROR DO error1
```

كما يمكن كتابة الأمر التالى :

```
ON ESCAPE EXIT
```

وذلك للخروج من حلقة تكرارية مثلا فى حالة ضغط المستخدم على مفتـاح الهروب (ESC) . كما يمـكن ايقـاف تأثـير مفـتاح الهـروب عن طريق كتابة الأمر (ON ESCAPE) دون كتابة أى شيء بعده .

٧٢ - الأمـر (PACK)

يستخدم هذا الأمر فى مسح السجلات التى سبق وضع علامات عليها تمهيدا لمسحها من الملف . والصورة العامة له كالآتى :

```
PACK
```

٧٣ - الأمـــر (PARAMETERS)

ويستخدم هذا الأمر فى تخصيص معاملات فى البرنامج الفرعى يتم من خلالها استقبال القيم التى يتـــم ادخالهـــا عنــــد استـــــدعاء هــــذا البرنامج الفرعى . وعند استدعاء هذا البرنامــــج الفرعـــــى يستخـــدم الأمـــــــر (DO <filename> WITH <parameter list>) حيث يتم ادخال القيم المقابلة للمتغيرات التى سبق تخصيصها . والصـــورة العامة لهذا الأمر كالآتى :

```
PARAMETERS <parameter list>
```

ويكتب هذا الأمر فى بداية البرنامج الفرعى سواء كان برنامج خطوات (Procedure) أو أى برنامج فرعى آخر .

وقائمة المعاملات (Parameter list) هي المتغيرات التي تستقبل المعاملات التى يتم امرارها مع الأمر (DO) . ويجب أن يكون عدد المعاملات فى أمر الاستقبال (PARAMETERS) وأمر الامرار (DO) واحدا .

واذا كان المعامل الذى يتم ادخاله او امراره (Passing) متغير ذاكرة ، فان قيمة هذا المتغير قد تتغير بناء على تشغيل البرنامج الفرعى ، ثم يتم نقل هذه القيمة الى البرنامج الذى قام باستدعاء البرنامج الفرعى .

مـــثال

الأوامر التالية تمثل برنامج خطوات او اجراءات (Procedure) اسمه (Calc) :

```
PARAMETERS length , width , area
area = length * width
RETURN
```

ولتنفيذ هذا البرنامج مع امرار القيمه (7) للطول (Length) . والقيـمة (9) للعرض (width) يتم كتابة الأمر التالى :

```
DO Calc WITH 7, 9, area
```

وللسؤال عن القيمة المخزنة فى المتغير (area) يتم كتابة الأمر التالى :

```
? area
```

وفى هذه الحالة يظهر العدد (63) الذى يمثل نتيجة المعادلة الموجودة فى البرنامج (Calc) .

٧٤ - الأمر (PRIVATE)

يستخدم هذا الأمر عندما يراد استخدام متغيرات ذاكرة فى برنامج فرعى بحيث لاتتأثر بأى متغيرات أخرى تحمل نفس أسماء هذه المتغيرات وتكون موجودة فى البرنامج الرئيسي أو فى أى برامج أخرى غير البرامج التابعة لهذا البرنامج الفرعى . كما يستخدم هذا الأمر أيضا لتغيير حالة متغير الذاكرة من متغير عام (Public) الى متغير خاص (Private) .وهناك صورتان للامر كالآتى :

```
PRIVATE <memvar list>
PRIVATE ALL  [LIKE/EXCEPT <skeleton>]
```

ويستخدم الاختيار (LIKE) عندما يراد استخدام الحروف الشاملة فى تغيير عدد من المتغيرات العامة (Public) الى خاصة (Private) . كما يستخدم الاختيار (EXCEPT) عندما يراد تحويل كل المتغيرات العامة الى متغيرات خاصة ما عدا بعض المتغيرات التى يتم تحديدها عن طريق استخدام الحروف الشاملة (Global Characters) .

فمثلا لتحويل كل المتغيرات التى تبدأ بالحرف (m) الى متغيرات خاصة يستخدم الأمر التالى :

```
PRIVATE ALL LIKE m*
```

كما يمكن تحويل كل المتغيرات الى متغيرات خاصة ماعدا المتغيرات التى تبدأ بالحرف (s) كالآتى :

```
PRIVATE ALL EXCEPT s*
```

ويجب ملاحظة أن المتغيرات الخاصة تختفى القيم الموجودة بها بمجرد انتهاء البرنامج ، وذلك عكس المتغيرات العامة (Public) التى تظل محتفظة بآخر قيمة تم ادخالها فيها .

٧٥ - الأمر (PROCEDURE)

يستخدم هذا الأمر لتحديد بداية كل برنامج خطوات او اجراءات

(Procedure) داخل ملف الخطوات (Procedure file) . والصورة العامة له كالآتى :

```
PROCEDURE <procedure name>
```

وكل برنامج خطوات او اجراءات (Procedure) يبدأ بالأمر (PROCEDURE) يليه اسم برنامج الخطوات . واسم برنامج الخطوات يتكون من ٨ حروف . ويمكن أن يحتوى على حروف أو أرقام أو شرطة سفلية (Underscore) . كما يجب أن يبدأ بحرف (Character).

مثــال

يمكن أن يحتوى ملف الخطوات او الاجراءات (Proc1) على برامج الخطوات الآتية :

```
PROCEDURE message
      @ 10,0 CLEAR
      @ 15,0 SAY "This is an invalid data"
RETURN
PROCEDURE printer
      @ 10,0 CLEAR
      @ 15,0 SAY "Send reprot to printer? Y/N" GET Pr;
           PICTURE"!"
        IF Pr = Y
           SET PRINT ON
        ENDIF
RETURN
```

ولتشغيل برنامج الخطوات (Printer) مثلا ، يتم أولا فتح ملف الخطوات (Proc1) وذلك كالآتى :

```
SET PROCEDURE TO Proc1
DO Printer
```

٧٦ - الأمــر (PUBLIC)

يستخدم هذا الأمر فى تحويل المتغيرات الى متغيرات عامة يمكن استخدامها فى أى برنامج فرعى أو رئيسى . وهذه المتغيرات تختلف عن باقى متغيرات الذاكرة فى أن القيم المخزنة بها تظل موجودة حتى بعد انتهاء البرنامج ، وتظل موجودة فى الذاكرة

المؤقتة حتى يتم مسحها باستخدام الأمر (RELEASE) أو (CLEAR MEMORY) . والصورة العامة للأمر كالآتي :

```
PUBLIC <memory variable list>
```

وعند استخدام هذا الأمر فان أي برنامج فرعي أو رئيسي يستخدم هذه المتغيرات يمكنه تغيير القيم الموجودة فيها . ويجب ملاحظة أن المتغيرات العامة يجب اعلانها عامة (Declaring) قبل اعطائها أي قيمة . فمثلا الأوامر التالية تعتبر خطأ :

```
Answer = "Y"
PUBLIC Answer
```

وذلك لأن المتغير (Answer) قد تم اعطاؤه القيمة "Y" قبل اعلانه عاما .

٧٧ - الأمر (QUIT)

يستخدم هذا الأمر في اغلاق جميع الملفات والخروج من البرنامج والرجوع الى نظام التشغيل . والصورة العامة له كالآتي :

```
QUIT
```

ويساعد هذا الأمر على الخروج من البرنامج بطريقة آمنة دون تحطيم اى ملفات ، حيث أن اغلاق جهاز الحاسب دون استخدام هذا الأمر قد يؤدى الى تحطيم الملفات المفتوحة وفقد البيانات المخزنة فيها .

٧٨ - الأمر (READ)

يستخدم هذا الأمر فى تخزين البيانات التى يتم ادخالها عن طريق الأمر(@...GET) . وهو يستخدم عادة فى تصميم شاشات الادخال من خلال البرنامج . والصورة العامة له كالآتي :

```
READ  [SAVE]
```

والاختيار (SAVE) يستخدم للابقاء على البيانات مخزنة في الـ (GETS) حتى يتم مسحها بواسطة الأمر (CLEAR GETS) ، حيث أن الوضع الطبيعي أن يتم مسح هذه الـ (GETS) بمجرد ادخال البيانات فى متغيرات الذاكرة ، فمثلا عند كتابة الأوامر التالية :

```
STORE "   " TO mname
@ 10,10 SAY "Enter your name:" GET mname
READ
```

فى هذه الحالة يتم تخزين الاسم الذى يدخله المستخدم في مخزن مؤقت (GET) . وعندما يقابل البرنامج الأمر (READ) فانه ينقل هذا الاسم الى متغير الذاكرة (mname) ، وبالتالى يتم اخلاء المخزن المؤقت (GET) .

أما عند استخدام الأمر (READ SAVE) بدلا من الأمر (READ) فى المثال السابق ، فان المخزن المؤقت (GET) يظل محتفظا بالاسم الذى تم ادخاله ، ولذلك يراعي عند استخدام الأمر (READ SAVE) عدة مرات التأكد من مسح المخازن المؤقتة (GETS) قبل أن يصل عددها الى الحد الأقصى المسموح به فى البرنامج الذى يكون (١٢٨) مخزنا فى الوضع المبدئى (Default) .

ويستخدم الأمر (READ) أيضا عند تصميم شاشات الادخال التى تتضمن أكثر من شاشة ، حيث يتم كتابة الأمر (READ) فى نهاية كل شاشة ويصبح فى هذه الحالة ناقل للصفحة التالية (Page break) . ويستطيع المستخدم في هذه الحالة الانتقال من أى شاشة الى الشاشة الأخرى باستخدام مفتاحى (PgDn) ، (PgUp) .

٧٩ - الأمر (RECALL)

يستخدم هذا الأمر في استعادة السجلات التي سبق وضع علامات عليها تمهيدا لمسحها باستخدام الأمر (PACK) . والصورة العامة له كالآتى :

```
RECALL [<scope>][WHILE <condition>][FOR <condition>]
```

والاختيارات كلها اختيارية فى هذه الحالة ، حيث يمكن كتابة الأمر دون كتابة أى شيء بعده ، وفى هذه الحالة يتم استعادة أول سجل فقط من السجلات التى سبق وضع علامات عليها لمسحها . وتستخدم الاختيارات فى استعادة السجلات التى تحقق شروطا معينة .

فمثلا لاستعادة أول سجل تم تجهيزه للمسح ، يتم كتابة السطر التالى :

```
RECALL
```

واذا أريد استعادة السجل رقم (١٠) يتم كتابة السطر التالى :

```
RECALL RECORD 10
```

واذا أريد استعادة جميع السجلات التى تم تجهيزها للمسح ، يتم كتابة السطر التالى :

```
RECALL ALL
```

حيث تمثل (ALL) المدى (scope) الذى يشمل كل السجلات التى تم تجهيزها للمسح .

٨٠ - الأمر (REINDEX)

يستخدم هذا الأمر فى اعادة انشاء ملف الفهرس المفتوح ، وذلك حتى يتم تحديث هذا الملف فى بعض الحالات التى تتطلب ذلك . والصورة العامة له كالآتى :

```
REINDEX
```

ويجب مراعاة فتح ملف الفهرس المراد تحديثه قبل كتابة هذا الأمر .

٨١ - الأمر (RELEASE)

يستخدم هذا الأمر فى مسح متغيرات الذاكرة (Memory Variables) مما يتيح استخدام الذاكرة المتاحة فى ادخال متغيرات أخرى .

والصورة العامة له كالآتى :

```
RELEASE <memvar list> [ALL][LIKE/EXCEPT <skeleton>]
        [MODULE <module name>]
```

ويستخدم الاختيار (skeleton) عندما يراد استخدام الحروف الشاملة

(Global) في تحديد المتغيرات المطلوب مسحها أو استثناء بعض المتغيرات من المسح حسب الحاجة .

كما يستخدم الاختيار (LIKE) لتحديد المتغيرات المشتركة في حرف معين أو عدة حروف .

ويستخدم الاختيار (EXCEPT) في استثناء المتغيرات المشتركة في حرف معين أو عدة حروف .

كما يستخدم الاختيار (ALL) في مسح جميع متغيرات الذاكرة .

ويستخــدم الاختيار (MODULE) في مسح برنامج مكتوب بلغة التجميع (Assembly) من الذاكرة ، ويساعد هذا على استخدام عدة برامج خارجية تزيد عن الحد الأقصى لعدد هذه البرامج وهو خمسة .

أمـثلة

لمسح كل متغيرات الذاكرة التى تبدأ بالحرف (m) يتم كتابة السطر التالى :

```
RELEASE ALL LIKE m*
```

ولمسح جميع متغيرات الذاكرة ما عدا المتغيرات التى تبدأ بحرف (s) يتم كتابة السطر التالى :

```
RELEASE ALL EXCEPT s*
```

كما يمكن مسح متغيرات ذاكرة معينة بكتابة السطر التالى مثلا :

```
RELEASE mname , maddress , mage
```

٨٢ - الأمـر (RENAME)

يستخدم هذا الأمر في تغيير اسم ملف سبق تخزينه على القرص الى اسم جديد . والصورة العامة لهذا الأمر كالآتى :

```
RENAME <old filename> TO <new filename>
```

ويجب أن يتضمن الاسم القديم والاسم الجديد الامتداد ورمز وحدة الأقراص المخزن عليها الملف اذا كانت غير وحدة الأقراص الحالية (Current drive). وفى حالة تغيير اسم ملف قاعدة بيانات يحتوى على حقول ملاحظات (memo fields) يتم تغيير اسم ملف الملاحظات منفصلا .

مثال

لتغيير اسم الملف (Cadets) الى الاسم (Grades) يتم كتابة السطر التالى :

```
RENAME Cadets.dbf TO Grades.dbf
```

٨٣ - الأمر (REPLACE)

يستخدم هذا الأمر فى استبدال محتويات حقول معينة بقيم جديدة يتم ادخالها . ويتم الحصول على هذه القيم عادة من متغيرات ذاكرة (Memory Variables). والصورة العامة لهذا الأمر كالآتى :

```
REPLACE[<scope>]<field1>
   WITH <exp>[,<field2> WITH <exp>]
   [WHILE <condition>][FOR <condition>]
```

وفى حالة عدم ادخال مدى (scope) أو شروط معينة للبحث فان الاستبدال يتم على السجل الحالى (Current Record) . كما يتم تحديد الحقول التى يتم استبدالها عن طريق كتابة اسم كل حقل مكان الاختيارات (field1, field2, ...etc) . ويمكن استبدال جميع حقول السجل بقيم جديدة .

وفى حالة استبدال الحقل الفهرسى (Key Field) بقيم جديدة فان ذلك سوف يؤدى الى تحديث الفهرس (Updating) ، وبالتالى يحدث تغير فى ترتيب السجلات ، لذلك يراعى عند استبدال الحقل الفهرسى عدم استخدام المدى (scope) وأوامر البحث مثل (WHILE , FOR) لأن ترتيب السجلات يكون قد تغير .

مثال

لزيادة سعر الوحدة (Unit Cost) بمقدار (١٠ ٪) فى جميع السجلات ،

يتم كتابة السطر التالى :

```
REPLACE ALL Unitcost WITH Unitcost*1.1
```

٨٤ - الأمر (REPORT)

يستخدم هذا الأمر لعرض التقرير الذى سبق تصميمه متضمنا بيانات مجموعة من السجلات يتم تحديدها ، وقد يتم عرض التقرير على الشاشة أو طباعته على الطابعة أو تخزينه كملف آسكى (ASCII file) . والصورة العامة لهذا الأمر كالآتى :

```
REPORT FORM <filename>/?[<scope>] [WHILE <condition>]
 [FOR <condition>][PLAIN][HEADING <exp>][NOEJECT]
 [TO PRINT/TO FILE <filename>][SUMMARY]
```

ولا يلزم هنا كتابة الامتداد فى اسم الملف (filename) حيث أن البرنامج يضيف الامتداد (.fmt) آليا .

ويمكن تلخيص الاختيارات المختلفة كالآتى :

الاختيارات (scope) ، (WHILE) ، (FOR) تستخدم لتحديد مدى وشروط البحث التى يتم عن طريقها تحديد السجلات المطلوب عرض بياناتها .

والاختيار (PLAIN) يؤدى الى عرض تقرير عادى لايحتوى على أرقام الصفحات أو تاريخ اليوم الحالى .

والاختيار (HEADING) يستخدم فى اضافة سلسلة حرفية تمثل عنوانا معينا للتقرير يكتب فى كل صفحة .

والاختيار (NOEJECT) يستخدم لالغاء نقل الصفحة (Page Break) ، وهذا يتيح لمخطط البرامج التحكم فى مكان الانتقال الى الصفحة التالية وبالتالى يمكنه التحكم فى طول الصفحة (Page Length) .

الاختيار (TO PRINT) يستخدم عندما يراد طباعة التقرير .

الاختيار (TO FILE) يستخدم لتخزين التقرير كملف آسكى (ASCII file) ، وهذا يتيح التعامل معه من خلال برامج أخرى مثل برامج الجداول الالكترونية .

الاختيار (SUMMARY) يستخدم في الحصول على تقرير مختصر يحتوى فقط على تجميع البيانات العددية الموجودة فى الملف ، وفى حالة استخدام الاختيار (Groups) عند انشاء التقرير من خلال برنامج المساعد (Assistant) ، يتم تجميع البيانات الخاصة بكل مجموعة منفصلة . ويراعى فى هذه الحالة فهرسة الملف باستخدام الحقل المستخدم فى التجميع (Grouping) كحقل فهرسى (Key field) .

٨٥ - الأمر (RESTORE)

يستخدم هذا الأمر فى استرجاع متغيرات الذاكرة التى سبق تخزينها فى ملف ذاكرة (Memory file) . والصورة العامة له كالآتى :

```
RESTORE FROM <filename> [ADDITIVE]
```

حيث (filename) هو اسم ملف الذاكرة المطلوب استرجاعه .

والاختيار (ADDITIVE) يستخدم لاضافة المتغيرات الموجودة فى الملف الى المتغيرات الموجودة فى الذاكرة المؤقتة فى هذا الوقت ، حيث أن عدم استخدام هذا الاختيار يؤدى الى مسح جميع المتغيرات الموجودة فى الذاكرة لتحل محلها المتغيرات الموجودة فى الملف .

ويجب ملاحظة أن جميع المتغيرات التى يتم استرجاعها تصبح خاصة (Private) بصرف النظر عن حالتها داخل الملف . وعند استرجاع الملف من مشيرة النقطة(Dot Prompt) تصبح المتغيرات عامة (Public) .

٨٦ - الأمر (RESUME)

يستخدم هذا الأمر لاستكمال تنفيذ البرنامج بعد ايقافه بواسطة الأمر (SUSPEND) . والصورة العامة له كالآتى :

```
RESUME
```

ويؤدى هذا الأمر الى استكمال تنفيذ البرنامج من نفس المكان الذى توقف عنده عند استخدام الأمر (SUSPEND) . ويستخدم هذا عادة فى اختبار وتصحيح البرنامج (Testing and Debugging) .

٨٧ - الأمر (RETURN)

يستخدم هذا الأمر فى الرجوع من البرنامج الفرعى الى البرنامج الذى قام باستدعائه ، أو الرجوع الى مشيرة النقطة (Dot Prompt) فى حالة استدعائه من البرنامج الرئيسى . والصورة العامة له كالآتى :

```
RETURN   [TO MASTER]
```

والاختيار (TO MASTER) يستخدم للرجوع الى البرنامج الرئيسى مباشرة من أى برنامج فرعى . وعند الرجوع الى البرنامج القائم باستدعاء البرنامج الفرعى يتم استكمال تنفيذ الأوامر التى تلى الأمر الذى قام باستدعائه .

واستخـــدام الأمر (RETURN) فى نهاية البرنامج يؤدى الى اغلاق هذا البرنامج، كما يؤدى الى مسح متغيرات الذاكرة الخاصة (Private) ولكنه لايؤثر فى المتغيرات العامة (Public) .

٨٨ - الأمـــر (RUN/!)

يستخدم هذا الأمر فى تشغيل أمر من أوامر نظام التشغيل (MS-DOS) من خلال

برنامج (DBase III+) . والصورة العامة له كالآتى :

```
RUN <command>
```

كما يمكن كتابته كالآتى :

```
! <command>
```

حيث يعمل الحرف (!) نفس عمل الأمر (RUN) .

ويمـــكن من خــــلال هـــذا الأمر تشغيل أوامــــر نظـــــام التشغيـــــل مثـــــل (DIR, RENAME , ERASE) .

ويجب ملاحظة أن هذا الأمر يتطلب ذاكرة مؤقتة أكبر من (٢٥٦) حرفا .

٨٩ - الأمر (SAVE)

يستخدم هذا الأمر فى تخزين متغيرات الذاكرة الموجودة فى الذاكرة المؤقتة فى ملف ذاكرة (Memory File) . والصورة العامة له كالآتى :

SAVE TO <filename> [ALL LIKE/EXCEPT <skeleton>]

حيث (filename) هو اسم الملف المطلوب تخزين متغيرات الذاكرة فيه . ويجب أن يحتوى اسم الملف على رمز وحدة الأقراص المراد تخزين الملف فيها اذا كانت غير وحدة الأقراص الحالية (Current Drive) . ولايتم كتابة الامتداد حيث أن البرنامج يضيف الامتداد (.mem) آليا .

والاختيار (ALL LIKE) يستخدم عندما يراد استخدام الحروف الشاملة فى تحديد مجموعة معينة من متغيرات الذاكرة التى تحتوى أسماؤها على حروف معينة .

والاختيار (ALL EXCEPT) يستخدم عندما يراد استثناء بعض المتغيرات التى تحتوى على حروف معينة من التخزين فى الملف .

والاختيار (skeleton) يستخدم لادخال الحروف الشاملة (Global Characters) المطلوب استخدامها .

وهناك حرفان يستخدمان كحروف شاملة وهما الحرف (*) وهو يحل محل أى عدد من الحروف والحرف (?) ويحل محل حرف واحد فقط .

أمثلة

عندما يراد تخزين جميع متغيرات الذاكرة الموجودة فى الذاكرة المؤقتة فى ملف اسمه (File1) مثلا يتم كتابة السطر التالى :

```
SAVE TO B: file1
```

ولتخزين جميع متغيرات الذاكرة الموجودة في الذاكرة المؤقتة والتى تبدأ بالحرف (m) فى الملف (File2) الموجود على وحدة الأقراص (B) يتم كتابة السطر التالى :

```
SAVE ALL LIKE m* TO B: file2
```

ويجب ملاحظة أنه عند تخزين متغيرات ذاكرة فى ملف سبق انشاؤه تظهر الرسالة التالية :

file2.mem already exists, overwrite it? (Y/N)

وعند كتابة (Y) فان المتغيرات الجديدة تمسح أى متغيرات أخرى سبق تخزينها فى الملف .

ولتخزين جميع المتغيرات التى لا تحتوى على الحرف (A) كحرف ثان فى اسم المتغير يتم كتابة السطر التالى :

```
SAVE ALL EXCEPT ?A*
```

حيث يمثل الحرف الشامل (?) الحرف الأول من اسم المتغير ، بينما يمثل الحرف الشامل (*) الحروف الباقية من اسم متغير الذاكرة .

٩٠ - الأمـر (SEEK)

يستخدم هذا الأمر فى البحث خلال ملف قاعدة البيانات المفهرس (Indexed) عن السجل الذى تطابق محتويات الحقل الفهرسى له قيمة معينة يتم ادخالها . والصورة العامة له كالآتى :

```
SEEK <expression>
```

والتعبير (expression) قد يكون قيمة عددية أو حرفية أو تاريخية أو علاقة حسابية . واذا كان قيمة حرفية (string) فيجب كتابتها بين علامات تنصيص (Quotation) . وهو فى هذا يختلف عن الامر الآخر (FIND) الذى لايحتاج الى وضع القيمة الحرفية المطلوب البحث عنها بين علامات تنصيص .

ويجب ملاحظة أن مقارنة القيمة الحرفية (String) تبدأ من أول حرف وحتى نهاية السلسلة (String) . فاذا كانت حروف السلسلة مطابقة لأول حروف فى الحقل الفهرسى ، فان الحقل يعتبر مطابقا بصرف النظر عن باقى حروفه ، وفى هذه الحالة يقف مؤشر السجلات (Record Pointer) عند أول سجل مطابق . واذا لم يتطابق الحقل مع القيمة الحرفية المطلوبة تظهر الرسالة "No found" . وفى هذه الحالة ينتقل المؤشر الى نهاية الملف .

وللبحث عن تاريخ معين يجب أولا تحويله من حروف (Characters) الى تاريخ (Date) وذلك باستخدام الدالة (CTOD) كالآتى مثلا :

```
SEEK CTOD ('01/01/90')
```

٩١ - الأمــر (SELECT)

يستخدم هذا الأمر فى التعامل مع مناطق العمل المختلفة (Work Areas) حيث يمكن عن طريق هذا الأمر فتح حتى عشر مناطق عمل ، وهذا يتيح التعامل مع عشرة ملفات قواعد بيانات فى نفس الوقت . والصورة العامة لهذا الأمر كالآتى :

```
SELECT <work area/alias>
```

ومناطق العمل (Work Areas) تأخذ الأرقام من (١) الى (١٠) أو الحروف من (A) الى (J) . كما يمكن استخدام المرادفات (Aliases) فى تسمية مناطق العمل والملفات الموجودة فيها . ويراعى عند اختيار المرادفات ألا تكون الحروف من (A) الى (J) . وذلك لأن هذه الحروف تمثل المرادفات المبدئية (Default) لمناطق العمل من (١) الى (١٠) . ولكن يمكن الجمع بين الحروف لتكوين مرادفات مثل (AA) أو (BB) وهكذا .

ويمكن فتح عدة ملفات قواعد بيانات فى مناطق عمل مختلفة . ولكن لايتم التعامل الا مع آخر منطقة عمل تم فتحها باستخدام الأمر (SELECT) .

ويمكن عرض بيانات من مناطق عمل أخرى غير منطقة العمل الحالية (Current Work Area) ، وذلك باستخدام العلامة (->) . وهذه العلامة تتكون من حرف الناقص (-) يليه حرف أكبر من (>) . فعند كتابة هذه العلامة بعد اسم منطقة العمل أو المرادف (Alias) ، يتم الحصول على البيانات الموجودة فى الحقل الخاص بالسجل الذى يقف عنده المؤشر (Pointer) . وذلك كالآتى :

```
Alias -> fieldname
```

ويتم ذلك عادة بعد ربط الملفات الموجودة فى مناطق عمل مختلفة باستخدام الأمر (SET RELATION) . ويجب ملاحظة أن كل منطقة عمل لها مؤشر سجلات (Record Pointer) خاص بها . كما أن مؤشر السجلات الوحيد الذى يمكن التحكم فيه هو مؤشر منطقة العمل الحالية (Current Work area) أما باقى المؤشرات فتظل ساكنة . وذلك باستثناء استخدام الأمر (SET RELATION) ، فانه يؤدى الى ربط جميع مؤشرات مناطق العمل المفتوحة ، بالمؤشر الموجودة فى منطقة العمل الحالية .

أمثلة

عندما يراد ربط ملف الطلبة (Cadets1) بملف الطلبة (Cadets2) يتم فتح

كل ملف فى منطقة عمل مختلفة كالآتى :

```
SELECT 1
USE Cadets1
SELECT 2
USE Cadets2 INDEX Name
SELECT 1
SET RELATION TO name INTO cadets2
```

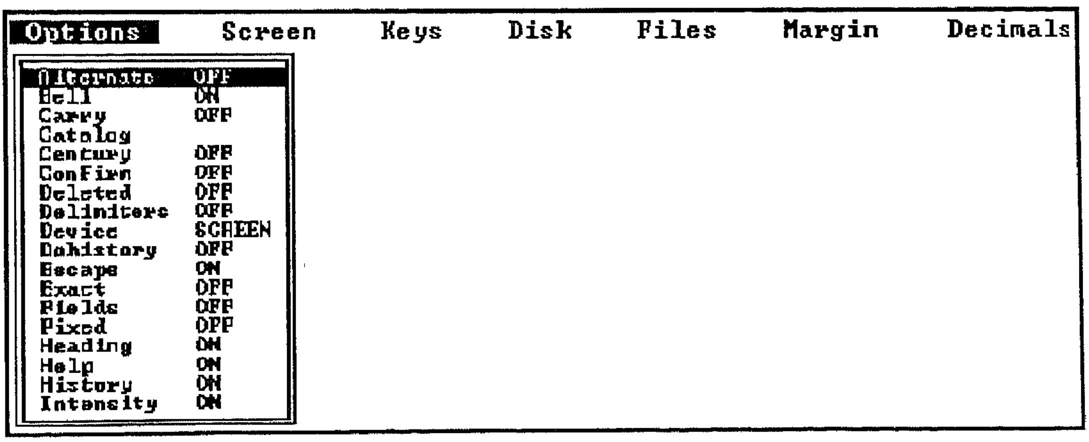
ويلاحظ فى هـذه الحالة أن منطقة العمل الحالية (Current) هـى المنطقة (١) . كما أن الملف الذى يتم ربطه بالملف الموجود فى منطقة العمل الحالية يجب أن يكون مفهرسا بناء على حقل مشترك بين الملفين .

٩٢ - الأمـر (SET)

يستخدم هذا الأمر فى عرض قائمة تتضمن بعض الاختيارات التى تساعد المستخدم على التحكم فى بيئة الحاسب (Environment) والصورة العامة له كالآتى :

```
SET
```

وعند كتابة هذا الأمر أمام مشيرة النقطة يظهر عمود الاختيارات الذى يتضمن الاختيارات الآتية : انظر الشكل (٢٧ - ١٩)



شكل (٢٧ - ١٩)

أ - الاختيار (Options)

ويحتوى هذا الاختيار على اختيارين . الاختيار الأول هو المساعدة (Help) وهو يؤدى الى عرض شاشات المساعدة التى تظهر مع بعض الأوامر ، وتقوم بارشاد المستخدم الى وظائف المفاتيح المختلفة . ويمكن كتابة الأمر الذى يؤدى هذه الوظيفة من مشيرة النقطة (Dot Prompt) ويكون كالآتى : (SET HELP ON) . كما يمكن ايقاف هذه الشاشات عن طريق كتابة الأمر (SET HELP OFF) .

والاختيار الثانى هو (Device) ويتم عن طريقه توجيه المخرجات الى الشاشة (Screen) أو الطابعة (Printer) . ويمكن كتابة الأمر الذى يؤدى هذه الوظيفة من مشيرة النقطة (Dot Prompt) . ويكون كالآتى (SET DEVICE TO SCREEN) عندما يراد توجيه المخرجات الى الشاشة ، أو كالآتى (SET DEVICE TO PRINT) عندما يراد توجيه المخرجات الى الطابعة .

ب - الاختيار (Screen)

ويستخدم هذا الاختيار للتحكم في ألوان الشاشة واختيار الألوان المطلوبة للخلفية (Background) والأعمدة الضوئية (Highlights) وذلك من خلال قائمة الاختيارات الفرعية الخاصة به . ويمكن التحكم فى الألوان أيضا عن طريق مشيرة النقطة (Dot Prompt) وذلك بكتابة الأمر (SET COLOR TO) .

جـ - الاختيار (Keys)

ويستخدم هذا الاختيار فى تغيير وظائف المفاتيح المساعدة (Function Keys) التى تظهر فى قائمة الاختيارات الفرعية والموضح بها وظائف هذه المفاتيح مع ملاحظة أن المفتاح (F1) يكون محجوزا للمساعدة (Help) . وباقى المفاتيح من (F2) الى (F10) يتم تغيير وظائفها حسب الحاجة . ويمكن استخدام عددا من الحروف حتى (٣٠) حرفا تمثل وظيفة كل مفتاح . ويمكن تنفيذ هذه العملية من مشيرة النقطة (Dot Prompt) عن طريق الأمر (SET FUNCTION) .

مثال

لتخصيص المفتاح (F8) للأمر (CLEAR) من خلال مشيرة النقطة يتم كتابة السطر التالى :

SET FUNCTION 8 To 'CLEAR'

د - الاختيار (Disk)

ويستخدم هذا الاختيار في معرفة وحدة الأقراص المبدئية (Default) وتحديد وحدة الأقراص التي يتعامل البرنامج مع الملفات الموجودة فيها . ويمكن تنفيذ هذه العملية من خلال مشيرة النقطة عن طريق الأمر (SET DEFAULT TO) .

هـ - الاختيا ر (Files)

ويستخدم هذا الاختيار لفتح ما يسمى بالملف البديل (Alternate File) وهو ملف نصوص (Text File) يستخدم في تخزين الأوامر التي يتم كتابتها من مشيرة النقطة (Dot Prompt) .

ويمكن فتح هذا الملف من مشيرة النقطة باستخدام الأمر (SET ALTERNATE TO) .

و - الاختيار (Margin)

يستخدم هذا الاختيار في ضبط الهامش الايسر في التقارير وكذلك في حقول الملاحظات (Memo Fields) . ويمكن تنفيذ نفس العملية من خلال مشيرة النقطة عن طريق الأمر (SET MARGIN TO) .

ز - الاختيار (Decimals)

ويستخدم هذا الاختيار في تحديد عدد الكسور العشرية المطلوب ظهورها في الأعداد . ويمكن تنفيذ نفس هذه العملية من خلال مشيرة النقطة عن طريق كتابة الأمر (SET DECIMALS TO) .

٩٣ - الأمر (SET BELL)

يستخدم هذا الأمر في التحكم في الصوت الذي يصدر عند ادخال أي بيانات خطأ، أو عند امتلاء العمود الضوئي الممثل للحقل بالحروف عند ادخال البيانات . والصورة العامة له كالآتي :

```
SET BELL ON/OFF
```

والوضع المبدئى (Default) هو ON .

٩٤ - الأمر (SET CATALOG)

يستخدم هذا الأمر فى فتح أو اغلاق ملف الكتالوج . وملف الكتالوج هو ملف يحتوى على جميع ملفات قواعد البيانات والملفات المرتبطة بها ، والتي يتم انشاؤها داخل هذا الكتالوج ، وهو يشبه استخدام الفهارس (Directories) والفهارس الفرعية(Subdirectories) فى نظام التشغيل (MS-DOS) . والصورة العامة لهذا الأمر كالآتى :

```
SET CATALOG ON/OFF
```

وعند كتابة الأمر (SET CATALOG ON) فان أى ملفات قواعد بيانات يتم انشاؤها تضاف الى الملفات المخزنة في الكتالوج المفتوح . أما عند كتابة الأمر (SET CATALOG OFF) ، فان أى ملفات جديدة يتم انشاؤها لاتضاف الى هذا الكتالوج .

ويجب ملاحظة أن الأمر (SET CATALOG OFF) لايغلق الكتالوج ولكنه يمنع اضافة أى ملفات جديدة اليه ، وذلك عكس الأمر (SET CATALOG TO) دون كتابة أى شيء بعد الأمر ، فان هذا يؤدى الى اغلاق ملف الكتالوج وبالتالى عدم القدرة على التعامل مع الملفات المخزنة فيه .

٩٥ - الأمر (SET CATALOG TO)

يستخدم هذا الأمر فى فتح كتالوج معين . وهو يختلف عن الأمر (SET CATALOG ON) فى أنه يفتح كتالوج محدد باسمه . أما الأمر (SET CATALOG ON) فانه يؤدى الى تجهيز الكتالوج المفتوح لاستقبال أى ملفات أخرى يتم اضافتها . والصورة العامة لهذا الأمر كالآتى :

```
SET CATALOG TO <filename>
```

والأمر بهذه الصورة يؤدى الى فتح الكتالوج الذى يحمل الاسم (filename) اذا كان موجودا ، واذا لم يكن موجودا يتم انشاء كتالوج جديد بهذا الاسم . واسم الملف يتم كتابته بدون الامتداد حيث أن البرنامج يضيف الامتداد (.cat) آليا .

ويستخدم الأمر (SET CATALOG TO) بدون أى شيء بعده فى اغلاق ملف الكتالوج المفتوح ، وفى هذه الحالة لايمكن التعامل مع الملفات الموجودة فى هذا الكتالوج . وذلك عكس الأمر (SET CATALOG OFF) الذى لايغلق ملف الكتالوج ولكنه يمنع اضافة أى ملفات اليه فقط .

ويمكن استخدام علامة الاستفهام مع هذا الأمر كالآتى :

```
SET CATALOG TO?
```

وفى هذه الحالة تظهر قائمة بأسماء ملفات الكتالوج الموجودة على القرص الحالى (Current Drive) .

وعند استخدام ملف الكتالوج تصبح منطقة العمل العاشرة محجوزة لهذا الكتالوج . وبذلك يصبح متاحا فقط تسع مناطق عمل (Work Areas) للتعامل مع الملفات من خلالها .

٩٦ - الأمر (SET CENTURY)

يستخدم هذا الأمر فى عرض التاريخ أو ادخاله متضمنا الأرقام الدالة على القرن مثل (1955) مثلا بدلا من (55) فقط .

والصورة العامة له كالآتى :

```
SET CENTURY ON/OFF
```

والوضع المبدئى لهذا الأمر هو (OFF) أى عدم ظهور الأرقام الدالة على القرن .

ويجب ملاحظة أن عرض هذه الأرقام الاضافية لايؤثر على حجم الحقل الممثل للتاريخ ، حيث أن حجم حقل التاريخ يكون دائما ثمانية حروف (Bytes) .

مثال

```
. ? DATE()
  15/02/90
. SET CENTURY ON
. ? DATE()
  15/02/1990
```

٩٧ - الأمر (SET COLOR)

يستخدم هذا الأمر فى مساعدة مخطط البرامج على التحكم فى ألوان الشاشة . وهذا الأمر له صورتان يمكن تلخيصهما كالآتى :

أ- الصورة الأولى

```
SET COLOR ON/OFF
```

وتستخدم هذه الصورة فى التحويل بين الشاشة الملونة (Colored Monitor) والشاشة الأحادية اللون (Monochrom Monitor) . والوضع المبدئى لهذا الأمر هو الوضع الذى يتم تشغيل برنامج (+DBase III) من خلاله . فاذا بدأ التشغيل على شاشة أحادية اللون (Monochrom) ، يصبح الوضع المبدئى للأمر هو (OFF) . واذا بدأ التشغيل على شاشة ملونة يصبح الوضع المبدئى للأمر هو (ON) .

ب - الصورة الثانية

```
SET COLOR TO [<standard>][,<enhanced>][,<border>]
     [,<background>]
```

والاختيار (standard) يقصد به لون الشاشة الخارجية (standard display) .

والاختيار (enhanced) المقصود به لون أى أعمدة ضوئية (Highlights) تظهر على الشاشة .

والاختيار (Border) المقصود به الحدود الخارجية للشاشة .

والاختيار (Background) المقصود به لون الخلفية التى تظهر خلف الحروف . والألوان فى كل حالة من هذه الحالات يتم تمثيلها بحروف يمكن تلخيصها فى الجدول التالى :

اللون	الحرف	اللون	الحرف
أسود	N	أحمر	R
أزرق	B	بنفسجى	RB
أخضر	G	بني	GR
سماوى	BG	أبيض	W
فراغ	X		

وتستخدم علامة (*) مع أي لون لجعله يتلألأ (Blinking) . كما تستخدم علامة الجمع (+) مع أى لون للحصول على اللـــون الأشد اضاءة (High Intensity). فمثلاً للحصول على اللون الأصفر ، وهو غير موجود في الجدول ، يتم كتابة (GR+) وهو يعنى اللون البنى الشديد الاضاءة أى الأصفر .

مثـال

للحصول على شاشة تحتوى على حروف صفراء على خلفية حمراء على أن يكون لون الحروف داخل الأعمدة الضوئية أبيض على خلفية زرقاء مع حدود خضراء ، يتم كتابة السطر التالى :

```
SET COLOR TO GR+/R, W/B , G
```

٩٨ - الأمـر (SET CONFIRM)

يستخدم هذا الأمر فى شاشات الادخال ليساعد مخطط البرامج على التحكم فى انتقال المؤشر (Cursor) من حقل الى آخر ، حيث أن الوضع المبدئى (Default) أن ينتقل هذا المؤشر الى الحقل التالى بمجـــرد امتلاء الحقل بالحـــروف . فاذا أريد عدم انتقال المؤشر بعد امتلاء الحقل ، يستخدم هذا الأمر . وفى هذه الحالة لايتم الانتقال الى الحقل التالى الا بالضغط على مفتاح الادخال . والصورة العامة لهذا الأمر كالآتى :

```
SET CONFIRM ON/OFF
```

٩٩ - الأمـر (SET CONSOLE)

يستخدم هذا الأمر فى التحكم فى الشاشة من خلال البرنامج عن طريق فتحها أو

اغلاقها . والصورة العامة له كالآتى :

```
SET CONSOLE ON/OFF
```

ويفيد هذا عندما يراد طباعة التقارير دون ظهورها على الشاشة . ويجب مراعاة اعادة الأمر الى الوضع المبدئى (Default) وهو (ON) قبل نهاية البرنامج .

١٠٠ - الأمـر (SET DATE)

ويستخدم هذا الأمر فى تغيير صورة التاريخ حسب النظم المختلفة والصورة العامة له كالآتى :

```
SET DATE AMERICAN/ANSI/BRITISH/ITALIAN/FRENCH/GERMAN
```

والوضع المبدئى (Default) هو التاريخ الأمريكى (AMERICAN) . والطرق المختلفة لكتابة التاريخ يمكن تلخيصها كالاتى :

```
AMERICAN   = MM/DD/YY
ANSI       = YY.MM.DD
BRITISH    = DD/MM/YY
ITALIAN    = DD-MM-YY
FRENCH     = DD/MM/YY
GERMAN     = DD-MM-YY
```

مثـال

لادخال تاريخ معين فى متغير ذاكرة ثم تحويله الى الشكل الانجليزى (BRITISH) ، يتم كتابة السطور التالية :

```
.    mdate = CTOD('01/20/90')
.    SET DATE BRITISH
.    ? mdate
20/01/90
```

١٠١ - الأمـر (SET DEBUG ON)

يستخدم هذا الأمر عند اختبار البرنامج لاكتشاف الأخطاء التى قد تكون موجودة به . وهو يرسل خطوات تشغيل البرنامج كما ينفذها الحاسب الى الطابعة . والصورة العامة له كالآتى :

```
SET DEBUG ON/OFF
```

وعند كتابة الأمر (SET DEBUG ON) ، يتم توجيه مخرجات الأمر (SET ECHO ON) الى الطابعة بدلا من ظهورها على الشاشة .

١٠٢ - الأمـر (SET DECIMALS)

ويستخدم هذا الأمر لتحديد عدد الأرقام العشرية المطلوب عرضها بالنسبة للمدخلات العددية . والصورة العامة له كالآتى :

```
SET DECIMALS TO <numeric expression>
```

والوضع المبدئى (Default) هو ظهور رقمين عشريين فى العدد .

١٠٣ - الأمـر (SET DEFAULT TO)

يستخدم هذا الأمر للانتقال الى وحدة أقراص أخرى لتحميل ملفات أو برامج موجودة عليها . والصورة العامة له كالآتى :

```
SET DEFAULT TO <drive>
```

مثال

عندما يراد تحميل برنامج الطلبة (Cadets) من وحدة الأقراص (B) يتم كتابة السطور التالية :

```
SET DEFAULT TO B
USE Cadets
```

١٠٤ - الأمـر (SET DELETED)

يستخدم هذا الأمر ليتيح لمخطط البرامج التعامل مع السجلات التي تم وضع علامات عليها لمسحها ، حيث يمكنه عزل هذه السجلات بحيث لاتؤثر فيها الأوامر التي يتم ادخالها . والصورة العامة له كالآتي :

SET DELETED ON/OFF

وعند كتابة الأمر (SET DELETED ON) فان البرنامج يتعامل مع هذه السجلات كأنها غير موجودة .

١٠٥ - الأمـر (SET DEVICE)

يستخدم هذا الأمر في توجيه السطور التي يتم كتابتها بواسطة الأمر (@...SAY) الى الشاشة أو الطابعة حسب الحاجة . والصورة العامة له كالآتي :

SET DEVICE TO PRINT/SCREEN

والوضع المبدئي (DEFAULT) هو توجيه المخرجات الى الشاشة . وعند كتابة الأمر (SET DEVICE TO PRINT) فان مخرجات الأمر (@...SAY) يتم توجيهها الى الطابعة . أما مخرجات الأمر (@...GET) فانها لاتذهب الى الطابعة .

١٠٦ - الأمـر (SET DOHISTORY)

يستخدم هذا الأمر للتحكم في تخزين الأوامر في مخزن التاريخ (History) أو عدم تخزينها فيه حسب الحاجة . والصورة العامة له كالآتي :

SET DOHISTORY ON/OFF

ويجب ملاحظة أن هذا الأمر يؤثر فقط في الأوامر التي يتم كتابتها في البرنامج ، ولكنه لايؤثر في الأوامر المباشرة التي يتم ادخالها عن طريق مشيرة النقطة (Dot Prompt) ، حيث أن هذه الأوامر يتم تخزينها في مخزن التاريخ (History) دون الحاجة الى كتابة هذا الأمر . ويستخدم هذا الأمر عادة عند اختبار البرنامج وتصحيحه .

١٠٧ - الأمر (SET ECHO)

يستخدم هذا الأمر فى عرض أوامر البرنامج أثناء تنفيذها . والصورة العامة له كالآتى :

SET ECHO ON/OFF

والوضع المبدئى (Default) هو (Off) أى عدم ظهور الأوامر أثناء تنفيذها . وهو يستخدم عادة عند اختبار البرنامج وتصحيحه .

١٠٨ - الأمر (SET ESCAPE)

يستخدم هذا الأمر للتحكم في ايقاف البرنامج أو استمراره نتيجة للضغط على مفتاح الهروب (ESC) . والصورة العامة له كالآتى :

SET ESCAPE ON/OFF

والوضع المبدئي هو (ON) . وهذا يعنى أن الضغط على مفتاح الهروب يؤدى الى ايقاف تنفيذ البرنامج .

١٠٩ - الأمر (SET EXACT ON)

يستخدم هذا الأمر في التحكم فى مقارنة البيانات الحرفية حيث أن المقارنة في الوضع المبدئى (Default) تتم بين حروف السلسلة الحرفية الأولى والحروف المقابلة لها من السلسلة الحرفية الثانية ، فاذا انتهت السلسلة الأولى أصبحت السلسلتان متطابقتين بالرغم من أنهما فعليا قد يكونان غير متطابقتين تماما . ولذلك يستخدم هذا الأمر في تحويل المقارنة الى مقارنة تامة (Exact) أى تطابق السلسلتين تماما . والصورة العامة لهذا الأمر كالآتى :

SET EXACT ON/OFF

مثال

عند كتابة السطر التالى :

```
? 'abc' = 'abcdef'
```

يظهر على الشاشة النتيجة (.F.) أى غير صحيح (False) .

وعند كتابة السطر التالى :

```
? 'abcdef' = 'abc'
```

يظهر على الشاشة النتيجة (.T.) أى صحيح (True) ، وذلك لأن البرنامج يقارن بين السلسلتين حتى انتهاء السلسلة اليمنى .

وعند كتابة السطرين التاليين :

```
SET EXACT ON
? 'abcdef' = 'abc'
```

يظهر على الشاشة القيمة (.F.) أى غير صحيح (False) . وذلك لأن البرنامج يطابق السلسلتين مطابقة كاملة .

١١٠ - الأمر (SET FIELDS)

يستخدم هذا الأمر عندما يراد التحكم في الحقول التى سبق تحديدها بواسطة الأمر (SET FIELDS TO) ، حيث يتيح لمخطط البرامج استخدام هذه الحقول أو عدم استخدامها حسب الحاجة . والصورة العامة له كالآتى :

```
SET FIELDS ON/OFF
```

والوضع المبدئي (Default) هو (OFF) ، وهو يعنى أن جميع حقول قاعدة البيانات يمكن التعامل معها ، حتى لو سبق تحديد بعض الحقول باستخدام الأمر (SET FIELDS TO) . أما باستخدام الأمر (SET FIELDS ON) فان الحقول التى سبق تحديدها فقط هى التى يمكن استخدامها ، وفى هذه الحالة يمكن التعامل مع الحقول الموجودة فى منطقة العمل الحالية (Current work area) أو التعامل مع الحقول الموجودة فى مناطق العمل الأخرى فى نفس الوقت .

١١١ - الأمر (SET FIELDS TO)

يستخدم هذا الأمر في تحديد الحقول المطلوب التعامل معها فى ملف أو أكثر من ملفات قاعدة البيانات . والصورة العامة له كالآتى :

```
SET FIELDS TO  [<field list>/ALL]
```

وعندما يراد اضافة حقول من مناطق عمل أخرى غير منطقة العمل الحالية (Current work area) يستخدم الاسم المرادف (Aias) الخاص بهذه المنطقة يليه العلامة (->) التي يتم تكوينها من علامة الناقص (-) وعلامة أكبر من (>) ويلى ذلك اسم الحقل المطلوب ادخاله ضمن قائمة الحقول المطلوبة .

تحذير

عند استخدام الأمر (SET FIELDS ON) دون تحديد الحقول المطلوبة أولا باستخدام الأمر (SET FIELDS TO) ، لايمكن التعامل مع الحقول . ويلزم فى هذه الحالة استخدام الأمر (SET FIELDS OFF) للعودة الى الوضع المبدئى (Default) .

ملاحظة

كما سبق الايضاح يمكن استخدام الأمر (SET FIELDS TO) في التعامل مع حقول من ملفات فى عدة مناطق عمل فى نفس الوقت ولكن هذه الحقول لاتكون مرتبطة ببعضها ، أى لايمكن عرض بيانات سجل واحد مثلا يتكون من عدة حقول من مناطق عمل مختلفة .

والطريقة الوحيدة لربط هذه الحقول هى استخدام الأمر (SET RELATION TO) فى ربط الملفات مع بعضها .

١١٢ - الأمر (SET FILTER)

يستخدم هذا الأمر في ترشيح أو تصفية قاعدة البيانات بحيث تبدو كأنها لا تحتوى الا على السجلات التي تحقق الشروط الموجودة فى المرشح (Filter) . والصورة العامة لهذا الأمر كالآتى :

```
SET FILTER TO [FILE <filename> /?][<condition>]
```

ويمكن استخدام الأمر (SET FILTER TO) دون كتابة أي شيء بعده وفي هذه الحالة يتم اغلاق المرشح الذى سبق فتحه لملف قاعدة البيانات المفتوح .

والاختيار (FILE <filename>) يستخدم عندما يراد استخدام ملف البحث (Query File) الذى سبق انشاؤه بواسطة الأمر (CREATE/MODIFY QUERY) . وتستخدم علامة الاستفهام (?) بدلا من اسم الملف لعرض أسماء جميع ملفات البحث الخاصة بملف قاعدة البيانات المفتوح ، و ذلك لاختيار الملف المطلوب منها .

والاختيار (Condition) يستخدم فى ادخال الشرط أو الشروط المطلوب استخدامها فى ترشيح قاعدة البيانات .

مثال

اذا أريد التعامل مع السجلات الخاصة بالطلبة الذين التحقوا بالمعهد بعد (١٩٨٥/١/١) مثلا . يتم كتابة السطر التالى :

```
SET FILTER TO dat_ent > CTOD "01/01/85"
```

حيث (dat_ent) هو اسم الحقل الذى يمثل تاريخ التحاق الطالب فى المعهد . وفي هذه الحالة يتعامل البرنامج مع قاعدة البيانات كأنها لاتحتوى الا على بيانات الطلبة الذين يحققون هذا الشرط .

١١٣ - الأمر (SET FIXED)

يستخدم هذا الأمر عندما يراد عرض البيانات العددية محتوية على عدد ثابت من الأرقام العشرية . وهذا العدد يكون فى الوضع المبدئي رقمين ، وذلك فى حالة عدم استخدام الأمر (SET DECIMALS) . والصورة العامة لهذا الأمر هى :

```
SET FIXED ON / OFF
```

١١٤ - الأمـر (SET FORMAT)

يستخدم هـذا الأمر فـى فتـح ملـف التشكيـل (Format File) المطلـوب استخـدامـه فى ادخال البيانات الى ملف قاعدة البيانات . والصورة العامة لهذا الأمر كالآتى :

```
SET FORMAT TO [<filename>/?]
```

ويتم كتابة اسم الملف (filename) بدون الامتداد ، حيث أن البرنامج يفتـرض أن الامتداد (fmt.) فى هذه الحالة . ويمكن استخدام علامة الاستفهام (?) بدلا من اسم الملف لعرض أسماء ملفات التشكيل الخاصة بملف قاعدة البيانات المفتوح واختيار الملف المطلوب منها .

ويمكن اغلاق ملف التشكيل باستخـدام الأمر (CLOSE FORMAT) أو الأمـــر (SET FORMAT TO) دون كتابة أى شيء بعده .

١١٥ - الأمـر (SET FUNCTION)

يستخــدم هـــذا الأمـــر فـــى تخصـــيص وظائـــف لمفاتيح الوظائـف (Function Keys) . وفى هذه الحالة يتم تغيير وظيفة المفتاح السابقة بالوظيفة الجديـدة التـى يتـم كتابتها . والصورة العامة لهذا الأمر كالآتى :

```
SET FUNCTION <exp1> TO <exp2> [;]
```

حيث (exp1) هو الرقم المثل للمفتاح المطلوب تغيير وظيفته ، و (exp2) هو اسم الوظيفة المطلوب ادخالها ، وحرف الفاصلة المنقوطة (;) يمثل مفتاح الادخال ، وذلك لكى يتم تنفيذ الوظيفة المطلوبة بمجرد الضغط على المفتاح الخاص بها دون الحاجة الى الضغط على مفتاح الادخال .

ويمكن ادخال اسم الوظيفة المطلوبة (exp2) حتى (٣٠) حرفا . ويجب ملاحظة أنه يمكن تغيير وظائف المفاتيح العشرة ما عدا مفتاح (F1) لأنه يكون محجوزا لتشغيل شاشات المساعدة (Help) .

مثـال

لتخصيص المفتاح (F5) للأمر (DISPLAY) يتم كتابة السطر التالى :

```
SET FUNCTION 5 TO 'DISPLAY;'
```

ولتخصيص المفتاح (F8) لتنفيذ عدة وظائف ، يمكن كتابة السطر التالى :

```
SET FUNCTION 8 TO 'CLEAR;USE CADETS INDEX NAME;'
```

١١٦ - الأمر (SET HEADING)

يستخدم هذا الأمر عندما يراد التحكم فى عناوين الحقول التى تظهر مع الأوامر(DISPLAY, LIST, SUM, AVERAGE) ، وذلك بعرضها أو عدم عرضها حسب الحاجة . والصورة العامة لهذا الأمر كالآتى :

```
SET HEADING ON/OFF
```

والوضع المبدئى لهذا الأمر هو (ON)، وفى بعض الأحيان يريد مخطط البرامج ادخال العناوين التى يريدها باستخدام الأمر (@...SAY) ، فى هذه الحالة يقوم باستخدام الأمر (SET HEADING OFF) .

مثال

عندما يراد عرض سجل مثلا من قاعدة بيانات الطلبة يشمل الحقلين (name) ، (address) يتم كتابة السطور التالية :

```
USE Cadets
DISPLAY name, address
```

وعند الضغط على مفتاح الادخال يظهر الآتى :

```
Record #    name                address

1        Hatem Mahmoud      20 Tahreer street
```

واذا أريد عدم عرض عناوين الحقول تستخدم الأوامر التالية :

```
USE Cadets
SET HEADING OFF
DISPLAY name, address
```

وعند الضغط على مفتاح الادخال يظهر الآتى :

```
1      Hatem Mahmoud       20 Tahreer street
```

١١٧ - الأمـر (SET HISTORY)

يستخدم هذا الأمر فى تشغيل مخزن التاريخ (History) أو عدم تشغيله حسب الحاجة . والصورة العامة له كالآتى :

```
SET HISTORY ON/OFF
```

والوضع المبدئى لهذا الأمر هو (ON) . وهذا يسمح باسترجاع الأوامر التى سبق ادخالها من مشيرة النقطة (Dot Prompt) باستخدام مفتاح السهم لأعلى ومفتاح السهم لأسفل . وبالتالى يمكن تنفيذ الأمر المطلوب بالضغط على مفتاح الادخال فقط دون الحاجة الى كتابة هذا الأمر مرة أخرى .

١١٨ - الأمـر (SET HISTORY TO)

يستخـدم هـذا الأمر فى تحديد عدد الأوامر التى يمكن تخزينها فى مخزن التاريخ . والصورة العامة له كالآتى :

```
SET HISTORY TO <exp>
```

حيث (exp) هو عدد الأوامر المطلوب تحديده . والعدد المبدئى (Default) هو عشرون . والعدد المسموح بادخاله هو أى عدد من صفر الى ١٦ ألف أمر حسب الذاكرة المتاحة .

١١٩ - الأمـر (SET INDEX)

يستخدم هذا الأمر فى فتح ملفات الفهرس لاستخدامها فى ترتيب السجلات . والصورة العامة له كالآتى :

```
SET INDEX TO  [<list of index filenames>/?]
```

ويتم كتابة اسم الملف بدون الامتداد حيث أن البرنامج يضيف الامتداد (.ndx) آليا . ويمكن اغلاق ملفات الفهرس المفتوحة عن طريق كتابة الأمر (SET INDEX TO) دون كتابة أى شيء بعده . كما يمكن اغلاقها أيضا باستخدام الأمر (CLOSE INDEX) . وتستخدم علامة الاستفهام (?) بدلا من اسم الملف لعرض أسماء جميع ملفات الفهرس الخاصة بملف قاعدة البيانات المفتوح .

ويمكن فتح حتى سبعة ملفات فهرس فى نفس الوقت . ويجب ملاحظة أن هذا الأمر يفتح ملف الفهرس فقط ولكنه لاينشؤه . ويصبح أول ملف فهرسى يتم فتحه هو الملف الرئيسى (Master) . وعند اجراء أى تعديل فى بيانات ملف قاعدة البيانات ، يتم تحديث ملفات الفهرس المفتوحة تبعا لهذا التعديل .

مثال

عندما يراد فتح ملفات الفهرس الخاصة بملف الموظفين (Clients) يتم كتابة السطور التالية :

```
USE Clients
SET INDEX TO Job, Name
```

وهذا يعنى أن ملف الفهرس (Job) هو الملف الرئيسى (Master) .

١٢٠ - الأمر (SET MARGIN)

يستخدم هذا الأمر لضبط الهامش الشمال للتقارير المطبوعة . والصورة العامة له كالآتى :

```
SET MARGIN TO <exp>
```

حيث (exp) هو قيمة عددية تمثل هذا الهامش .

مثـال

```
SET MARGIN TO 10
```

١٢١ - الأمـر (SET MEMOWIDTH)

يستخدم هذا الأمر في تحديد عرض بيانات حقل الملاحظات (memo field) عندعرضها على الشاشة أو طباعتها . والصورة العامة له كالآتي :

```
SET MEMOWIDTH TO <exp>
```

والوضع المبدئي لهذا العرض هو (٥٠) حرفا .

١٢٢ - الأمـر (SET MENU)

يستخدم هذا الأمر للتحكم في ظهور أو عدم ظهور شاشات المساعدة التي تـظهر للمستخدم لتعريفه بوظائف مفاتيح التحكم في المؤشر لتصحيح المدخلات قبل ادخالها .

والصورة العامة لهذا لأمر كالآتي :

```
SET MENU ON/OFF
```

والوضع المبدئي هو (ON)

١٢٣ - الأمـر (SET ORDER)

يستخدم هذا الأمر في التحكم في ترتيب ملفات الفهرس المفتوحة وبالتالي تحديد أيها تصبح ملفا رئيسيا (Master) . والصورة العامة له كالآتي :

```
SET ORDER TO [<exp>]
```

حيث : (exp) هو رقم صحيح يأخذ القيمة من صفر الى (٧) حسب عدد ملفات الفهرس المفتوحة . وعند كتابة القيمة صفر يعود الملف الى وضعه الأول دون فهرسة . وذلك دون اغلاق ملفات الفهرس . وعند كتابة أي رقم آخر ، فان

هذا الرقم يمثل ترتيب ملف الفهرس الذى يصبح هو الملف الرئيسى (Master) .

مثال

لفتح ملفات الفهرس الخاصة بقاعدة بيانات الموظفين (Clients) يتم كتابة السطرين التاليين :

```
USE Clients
SET INDEX TO Job, Name
```

وعندما يراد استخدام ملف (Name.ndx) كفهرس رئيسى يتم كتابة السطر التالى :

```
SET ORDER TO 2
```

١٢٤ - الأمر (SET PATH)

يستخدم هذا الأمر فى تحديد مسار معين للملفات المطلوب استخدامها والصورة العامة له كالآتى :

```
SET PATH TO  [<path list>]
```

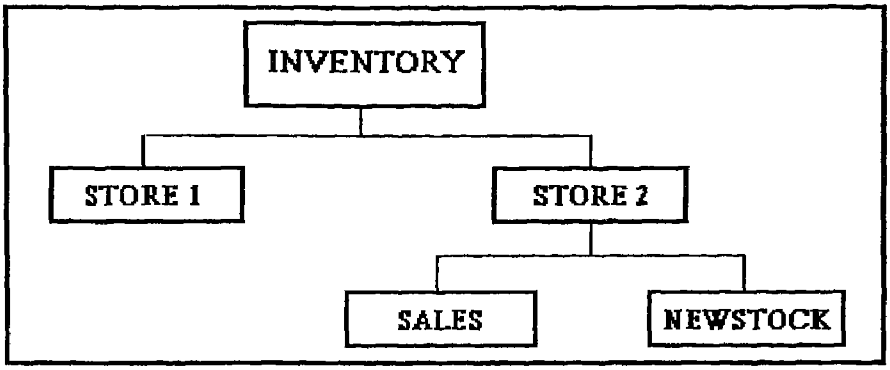
واستخدام الأمر (SET PATH TO) دون كتابة أى شىء بعده يعيد المسار الى الفهرس الفرعي الحالى (Current Directory) . وتحديد المسار يساعد البرنامج على البحث عن الملفات المطلوبة ، وذلك عن طريق تنظيمها على شكل شجرة(Tree) . حيث يبدأ البحث من الجذر (Root) وينتقل الى الفروع والفروع الأصغر وهكذا .

والمسار عبارة عن فهارس فرعية يفصلها الشرطة المائلة (\) ويبدأ بالفهرس الرئيسى (Root Directory) يليه الفهارس الفرعية الأخرى .

وقائمة المسارات (Path List) هى مجموعة من المسارات (Pathes) يفصل بينها فاصلة أو فاصلة منقوطة (;) . وعندما يبحث البرنامج عن أى ملف فانه يبحث

فى المسار الأول . فاذا لم يجده ، ينتقل الى المسار الثانى ثم الثالث ... وهكذا .

انظر الشكل (٢٧ - ٢٠)



الشكل (٢٧ - ٢٠)

مثال

اذا كان الملف (Cadets.prg) غير موجود فى الفهرس الحالى فان الأوامر التالية تؤدى الى البحث عنه فى مسارات أخرى كالآتى :

```
SET PATH TO A:\DBase, B:\Hasan
DO Cadets
```

وفى هذه الحالة يتم البحث فى الفهرس الفرعى (DBase\) أولا ، ثم يتم الانتقال الى الفهرس الفرعى (Hasan\) حتى يتم الوصول الى الملف المطلوب .

١٢٥ - الأمر (SET PRINT)

يستخدم هذا الأمر فى توجيه المخرجات التى لايتم كتابتها باستخدام الأمر (@...SAY) الى الطابعة مع ظهورها على الشاشة فى نفس الوقت ، وهى البيانات الناتجة من استخدام الأوامر (? , LIST , DISPLAY) . والصورة العامة له كالآتى :

```
SET PRINT ON/OFF
```

والوضع المبدئي له (OFF) .

١٢٦ - الأمر (SET PROCEDURE)

يستخدم هذا الأمر في فتح ملف الخطوات أو الاجراءات (Procedure File) . والصورة العامة له كالآتي :

```
SET PROCEDURE TO [<filename>]
```

واسم الملف يجب أن يتضمن رمز وحدة الأقراص اذا كانت غير وحدة الأقراص الحالية (Current Drive) . ولا يشترط اضافة الامتداد حيث أن البرنامج يضيف الامتداد (.prg) آليا .

ويتم اغلاق ملف الخطوات او الاجراءات عن طريق كتابة الأمر (SET PROCEDURE TO) دون كتابة أي شيء بعده .

وملف الخطوات يمكن أن يحتوى على (٣٢) برنامج خطوات (Procedure) . ويتم تحديد بداية كل برنامج عن طريق كتابة الأمر (PROCEDURE) في أوله . ويجب ملاحظة أنه لايمكن فتح أكثر من ملف خطوات في نفس الوقت . واذا أريد استخدام عدة ملفات ، يتم اغلاق الملف السابق وفتح ملف جديد وهكذا .

١٢٧ - الأمر (SET RELATION)

يستخدم هذا الأمر في ربط ملفين من ملفات قواعد البيانات باستخدام حقل مشترك بين الملفين . والصورة العامة له كالآتي :

```
SET RELATION TO[<keyfield>/<RECNO()> INTO <alias>]
```

ويقوم هذا الأمر بربط ملف قاعدة البيانات الموجود في منطقة العمل الحالية (Current Work Area) بملف آخر مفتوح في منطقة عمل أخرى . وهذا الملف الآخر يتم تحديده عن طريق الاختيار (alias) أو المرادف الذى يشمل اسم منطقة العمل واسم الملف المفتوح فيها .

ويمكن ربط الملفين عن طريق حقل مشترك فيهما (keyfield) ، وفي هذه الحالة يجب أن يكون الملف المربوط مفهرسا (Indexed) بناء على هذا الحقل المشترك .

كما يمكن ربط الملفين عن طريق رقم السجل (RECNO()) . وفي هذه الحالة يجب أن يكون الملف المربوط غير مفهرس (Unindexed) . ويمكن تخزين هذه العلاقة في ملف منظر (VIEW FILE) ، وذلك عن طريق الأمر (CREATE VIEW FROM ENVIRONMENT) .

مثال

اذا كان هناك ملفان لبيانات الطلبة (Cadets1) ، (Cadets2) ويراد ربطهما بناء على الحقل المشترك بينهما وهو حقل الاسم (name) يتم كتابة الأوامر التالية :

```
SELECT 1
USE Cadets1
SELECT 2
USE Cadets2 INDEX name
SELECT 1
SET RELATION TO name INTO cadets2
```

يلاحظ أن الملف المطلوب ربطه وهو (Cadets2) تم فهرسته بناء على الحقل المشترك بين الملفين وهو حقل الاسم (name) . كما يلاحظ أن آخر منطقة عمل تم اختيارها بواسطة الأمر (SELECT) هي منطقة العمل التي يتم انشاء العلاقة داخلها .

١٢٨ - الأمر (SET SAFETY)

يستخدم هذا الأمر في تأمين الملفات عن طريق منع النسخ فوقها ، ويحدث هذا عندما يريد المستخدم نسخ ملف باسم ملف موجود . في هذه الحالة تظهر الرسالة التالية :

```
<filename> already exists, overwrite it ? Y/N
```

وفي بعض الأحيان يريد مخطط البرامج التحكم في نسخ الملفات دون ظهور هذه

الرسالة التى تسبب توقف البرنامج ، ولذلك يقوم باستخدام هذا الأمر . والصورة العامة له كالآتى :

```
SET SAFETY ON/OFF
```

والوضع المبدئى لهذا الأمر هو (ON) .

١٢٩ - الأمـــر (SET SCOREBOARD)

يستخدم هذا الأمر عندما يراد استخدام السطر رقم صفر الذى يكون محجوزا عادة لرسائل برنامج (DBase III+) فــــى حالـــة استخــــدام الأمــــــر (SET STATUS OFF) . والصورة العامة له كالآتى :

```
SET SCOREBOARD ON/OFF
```

وهو يكون عادة (ON) .

١٣٠ - الأمـــر (SET STATUS OFF)

يستخدم هذا الأمر للتحكم فى ظهور عمود الحالة (Status Bar) أو عدم ظهوره . والصورة العامة له كالآتى :

```
SET STATUS ON/OFF
```

والوضع المبدئى له (ON) . ويتيح هذا الأمر لمخطط البرامج استخدام السطر رقم (٢٢) الذى يكون محجوزا لعمود الحالة فى عرض بيانات على الشاشة أو تصميم شاشات الادخال .

١٣١ - الأمـــر (SET STEP)

يستخدم هذا الأمر فى تنفيذ خطوات البرنامج خطوة خطوة مع الوقوف بعد كل خطوة ، ويساعد هذا على اكتشاف أخطاء البرنامج ان وجدت . والصورة العامة لهذا الأمر كالآتى :

```
SET STEP ON/OFF
```

والوضع المبدئى له (OFF) .

١٣٢ - الأمـــر (SET TALK)

يستخدم هذا الأمر فى التحكم فى عرض خطوات تنفيذ البرنامج على الشاشة أوعدم عرضها . والصورة العامة له كالآتى :

```
SET TALK ON/OFF
```

والوضع المبدئى له (ON) . وعادة يحوله مخطط البرامج الى (OFF) عن طريق كتابة الأمـر (SET TALK OFF) فى بداية البرنامج ، حتى يتحكم فيما يعرض على الشاشة أو على الطابعة .

مثـــال

يمكن ملاحظة الفرق بين الوضع (ON) والوضع (OFF) مع هذا الأمر من خلال السطور التالية :

```
.     STORE 'Mohamed' TO mname1
Mohamed
.     SET TALK OFF
.     STORE 'Hasan' TO mname2
```

يلاحظ بعد السطر الأول ظهور محتويات متغير الذاكرة (mname) نتيجة تنفيذ الأمر ، أما بعد السطر الأخير فلا تظهر محتويات متغير الذاكرة نتيجة استخـــدام الامر (SET TALK OFF).

١٣٣ - الأمـــر (SET TYPEAHEAD)

يستخدم هذا الأمر فى تحديد عدد الحروف المسموح بالاحتفاظ بها فى مخزن الذاكرة (Buffer) قبل انتقالها الى الذاكرة الداخلية . والصورة العامة له كالآتى :

```
SET TYPEAHEAD TO <exp>
```

حيث (exp) هو عدد الحروف المطلوب تحديده . والعدد المبدئي يكونعشرين حرفا . وهـذا الأمـر يعمـل فقط عندما يكون الأمر (SET ESCAPE) على الوضع (ON). أى أنه لا يعمل عند استخدام الأمر (SET ESCAPE OFF) . وكذلك لا يعمل عند استخدام أى أمر من الأوامر (ON KEY) ، (INKEY) حيث أن هذا يؤدى الى ادخال أى حرف يكتبه المستخدم الى الذاكرة الداخلية مباشرة .

١٢٤ - الأمـر (SET UNIQUE)

يستخدم هذا الأمر للتحكم فى ادخال السجلات التى تشترك فى نفس قيمة الحقل الفهرسى أو عدم ادخالها فى الفهرس . والصورة العامة له كالآتى :

```
SET UNIQUE ON/OFF
```

والوضع المبدئى له (OFF)

وعند انشاء ملف فهرس مع استخدام الأمر (SET UNIQUE ON) فان السجلات التى تحتوى على نفس القيمة للحقل الفهرسى لايتم ادخالها فى الفهرس ، ولكن يتم ادخال أول سجل فقط ، وفى هذه الحالة لايكون الملف الفهرسى محتويا على أرقام سجـلات مشتركـة فى قيمـة الحقل الفهرسى . واستخدام هذا الأمر قبل فتح ملف الفهرس (Index File) يؤدى نفس العمل الذى يؤديه الأمر:

```
INDEX ON <key expression> TO <filename> UNIQUE
```

وعنـد اضافـة أو تعديل أى سجلات فى الملف فان الملف يستعيد الحالة المنفردة (Unique) ، أى أن اضافة أى سجلات تحتوى على حقل فهرسى يماثل الحقل الفهرسى لسجلات أخرى موجودة فى الملف ، لايؤدى الى اضافة أرقام هذه السجلات فى الفهرس باعتبارها سجلات متكررة .

١٣٥ - الأمر (SET VIEW TO)

يستخدم هذا الأمر فى فتح ملف المنظر (View File) . والصورة العامة له كالآتى :

```
SET VIEW TO <filename>/?
```

ويتم ادخال اسم الملف (filename) بدون الامتداد حيث أن البرنامج يفترض الامتداد (.vue) ، وتستخدم علامة الاستفهام (?) لعرض أسماء جميع ملفات المنظر المخزنة على القرص أو فى الكتالوج المفتوح .

١٣٦ - الأمر (SKIP)

يستخدم هذا الأمر في نقل مؤشر السجلات (Record Pointer) الى سجل تال أوسجل سابق خلال ملف قاعدة البيانات المفتوح . والصورة العامة له كالآتى :

```
SKIP   [<exp>]
```

حيث (exp) هو عدد السجلات المطلوب تحريك المؤشر خلالها . والمؤشر يتحرك الى السجلات التالية بما يساوي هذا العدد . كما يمكن أن يتحرك الى السجلات السابقة عند كتابة علامة ناقص (-) قبل العدد (exp) . واذا لم يتم اضافة أي عدد الى الأمر ، فانه يؤدى الى تحريك المؤشر سجلا واحدا الى الأمام .

أمثلة

لتحريك المؤشر الى السجل الثانى فى ملف قاعدة بيانات الطلبة (Cadets) يتم كتابة الأوامر التالية :

```
USE Cadets
SKIP
```

ولتحريك المؤشر من السجل الثانى الى السجل رقم (٩) يتم كتابة السطر التالى :

```
SKIP 7
```

وللرجوع الى السجل رقم (٦) يتم كتابة السطر التالى :

```
SKIP -3
```

ويمكن تخزين عدد معين فى متغير ذاكرة . واستخدام متغير الذاكرة مع الأمر(SKIP) . وذلك كالآتى :

```
STORE 2 TO mskip
SKIP mskip
```

١٢٧ - الأمـــر (SORT)

يستخدم هذا الأمر فى انشاء ملف قاعدة بيانات يحتوى على سجلات الملف الأصلى مرتبة بالترتيب المطلوب تبعا لمحتويات حقل معين أو عدة حقول , والصورة العامة له كالآتى :

```
SORT [<scope>] TO [<filename>] ON <field1>[/A][/D][/C]
[,<field2>][/A][/D][/C] .... [WHILE <condition>]
 [FOR  <condition]
```

والاختيار (scope) يستخدم لتحديد المدى فى الملف المطلوب ترتيبه ، حيث يمكن ترتيب الملف كله (ALL) أو مجمـــوعة من السجلات ابتداء من سجل معين (REST) . والاختيار (filename) هو اسم ملف قاعدة البيانات المرتب الذى سيتم انشاؤه .

والاختيارات (field1) ، (field2) تستخدم لتحديد الحقل أو الحقول التى يتم الترتيب بناء عليها . ويمكن استخدام حتى عشرة حقول فى ترتيب الملف ، والاختيار (/A) يستخدم للترتيب تصاعديا (Ascendingly) . ويجب ملاحظة أن الوضع المبدئى هو الترتيب تصاعديا .

والاختيار (/D) يستخدم للترتيب تنازليا (Descendingly) والاختيار (/C) يستخدم عندما يراد عدم التمييز بين الحروف الكبيرة (Capital) أو الصغيرة (Small) فى الترتيب .

كما يستخدم الاختياران (WHILE) ، (FOR) فى تحديد شروط البحث التى يتم من خلالها اختيار السجلات المطلوب ترتيبها .

مثـال

يمكن ترتيب ملف الطلبة (Cadets) بناء على حقلى المهنة (Job) والاسم (name) كالآتى :

```
SORT ON Job, name TO scadets
```

حيث يصبح الملف (scadets) ملف قاعدة بيانات آخر يحتوى على نفس سجلات الملف الأصلى (Cadets) مرتبة حسب حقلى المهنة والاسم على الترتيب .

١٣٨ - الأمـر (STORE)

يستخـــدم هـــذا الأمــــــر فـــى انشاء متغير أو عدة متغـــيرات ذاكـرة (memory variables) والصورة العامة له كالآتى :

```
STORE <exp> TO <memory variable list>
```

حيث (exp) هى القيمة أو القيم التى يتم تخزينها فـــى متـغـــيرات الـذاكـرة . ويمكن تخزين قيمة واحدة فى عـــدة متغيرات يتم تحديدها مـــن خلال الاختيار (memory variable list) . وهناك طريقة أخرى يمكن عن طريقها انشاء متغير الذاكرة وذلك كالآتى :

```
<memory variable> = <expression>
```

ولكن هذه الطريقة لاتسمح بانشاء عـــدة متغـــيرات فى سطر واحـد مثل الأمر(STORE) .

ملاحـــظة

الحقل الذى يحمل نفس اسم متغير الذاكرة تكون له الأسبقية عند تنفيذ أى عملية على هذا الحقل . فمثلا اذا كان هناك حقل يحمل الاسم (name) ومتغير ذاكرة يحمل نفس الاسم (name) ، فان أى عملية يتم اجراؤها على متغير الذاكرة لاتؤثر فيه ولكنها تؤثر فى الحقل (name) فقط .

مثـــال

لتخزين القيمة صفر فى عدة متغيرات (a, b, c) يتم كتابة السطر التالى :

```
STORE 0 TO a, b, c
```

١٣٩ - الأمـر (SUM)

يستخدم هذا الأمر فى تجميع الحقول العددية فى ملف قاعدة البيانات المفتوح والصورة العامة له كالآتى :

```
SUM [<scope>][<fields>][TO <memvar list>]
  [WHILE <condition>][FOR <condition>]
```

ويلاحظ أن جميع الاختيارات اختيارية ، أى يمكن كتابة الأمر دون كتابة أى شيء بعده وفى هذه الحالة يتم تجميع جميع الحقول العددية لجميع سجلات قاعدة البيانات .

والاختيار (scope) يتم من خلاله تحديد المدى الذى يتم البحث فيه عن السجلات المطلوب تجميع بيانات الحقول العددية فيها .

والاختيار (fields) يستخدم لتحديد الحقول المطلوب تجميعها .

والاختيار (TO <memvar list>) يستخدم عندما يراد تخزين هذا المجموع فى متغيرات ذاكرة يتوقف عددها على عدد الحقول المطلوب تجميعها .

كما يستخدم الاختياران (WHILE) ، (FOR) للبحث عن السجلات المطلوب تجميع بياناتها العددية .

مثـــال

عندما يراد تجميع حقلى المرتب (Salary) ، والساعات (Hours) فى ملف الموظفين (Clients) يتم كتابة السطر التالى :

```
SUM salary , hours TO msalary , mhours
```

١٤٠ - الأمـر (SUSPEND)

يستخدم هذا الأمر فى ايقاف تنفيذ أوامر البرنامج ايقافا مؤقتا . ويمكن كتابة هذا الأمر داخل البرنامج ليساعد مخطط البرامج على ايقاف البرنامج فى المكان الذى يشك فى وجود خطأ فيه ، حيث يستطيع عرض الأوامر السابقة لهذا المكان والمخزنة فى

مخزن التاريخ (History) ، ثم يمكنه العودة مرة ثانية الى تنفيذ أوامرالبرنامج عن طريق الأمر (RESUME) .

١٤١ - الأمــر (TEXT)

يستخدم هذا الأمر فى كتابة نصوص كبيرة على الشاشة أو الطابعة . والصورة العامة له كالآتى :

```
TEXT
<text characters>
ENDTEXT
```

ويتم كتابة النص بين (TEXT) ، (ENDTEXT) ويمكن استخدام هذا الأمر فى عرض قوائم الاختيارات الكبيرة أو شاشات المساعدة (Help) .

مثــال

يمكن كتابة السطور التالية فى البرنامج :

```
TEXT
this is an example of text that is to be
displayed on the screen as it is.
ENDTEXT
```

١٤٢ - الأمــر (TOTAL)

يستخدم هذا الأمر فى تجميع بيانات الحقول العددية لكل السجلات أو لمجموعة من السجلات التي تحقق شروطا معينة ، ويتم التجميع لكل مجموعة من السجلات تشترك فى قيمة الحقل الذى يستخدم كمفتاح (Key Field) لتقسيم هذه المجموعات . والصورة العامة لهذا الأمر كالآتى :

```
TOTAL ON <keyfield> TO <filename>[<scope>]
  [FIELDS <fieldlist>][WHILE <condition>]
  [FOR <condition>]
```

والاختيار (keyfield) يستخدم في تحديد المجموعات التى يتم تجميع الحقول العددية بها . وهو أحد حقول ملف قاعدة البيانات .

واسم الملف (filename) هو ملف قاعدة بيانات جديد يتم انشاؤه متضمنا مجموع البيانات العددية للمجموعات المختلفة من السجلات .

والاختيار (scope) يستخدم لتحديد مدى السجلات المطلوب البحث خلاله عن السجلات المطلوب ادخالها فى الملف الجديد .

والاختيار (FIELDS <field list>) يستخدم لتحديد أسماء الحقول المطلوب ادخالها فى الملف الجديد .

ويستخدم الاختياران (WHILE) ، (FOR) فى تحديد السجلاتَ المطلوب ادخالها فى الملف الجديد حسب شروط البحث التى يتم ادخالها .

ويجب ملاحظة أن ملف قاعدة البيانات المفتوح والذى يتم تجميع بياناته يجب أن يكون مفهرسا (Indexed) أو مفروزا (Sorted) بناء على الحقل المستخدم كمفتاح (Key Field) .

ملاحظة

تركيب الملف الجديد (Structure) يكون مماثلا تماما لتركيب ملف قاعدة البيانات الأصلى باختلاف واحد وهو عدم نسخ حقول الملاحظات (memo fields) فى الملف الجديد .

مثال

لتجميع بيانات حقول ملف قاعدة بيانات الموظفين (Clients) تبعا لحقل المهنة (Job) يتم كتابة السطور التالية :

```
USE Clients
INDEX ON Job TO Job
TOTAL ON Job TO Job
```

ويلاحظ من السطر الثانى أنه تم استخدام الحقل (Job) كحقل فهرسى لانشاء ملف الفهرس (Job.ndx) .

وفى السطر الثالث تم تجميع بيانات ملف قاعدة البيانات المفتوح وهو الملف (Clients) بناء على حقل المهنة (Job) ، ونتيجة لذلك تم انشاء ملف قاعدة البيانات (Job.dbf) .

وعند عرض بيانات حقلى المرتب (Salary) وساعات العمل (Hours) فى ملف قاعدة البيانات الجديد ، يتم ذلك كالآتى :

```
USE Job
LIST salary , hours
```

يلاحظ فى هذه الحالة ظهور الآتى مثلا :

Record	Job	salary	hours
1	Engineer	256570.00	700
1	Technical	189500.00	650

ويلاحظ هنا مجموع مرتبات المهندسين وكذلك الفنيين .

١٤٣ - الأمـــر (TYPE)

يستخدم هذا الأمر فى عرض محتويات ملف نصوص (Text file) . والصورة العامة له كالآتى :

```
TYPE <filename>  [TO PRINT]
```

واسم الملف (Filename) يجب أن يتضمن الامتداد ورمز وحدة الأقراص المخزن عليها اذا كانت غير وحدة الأقراص الحالية (Current drive) .

١٤٤ - الأمـــر (UPDATE)

يستخدم هذا الأمر فى تحديث بيانات ملف قاعدة البيانات المفتوح من ملف آخر مفتوح فى منطقة عمل أخرى ، ويتم ادخال التعديلات بمطابقة الملفين تبعا لقيمة حقل معين يستخدم كمفتاح للمطابقة . والصورة العامة لهذا الأمر كالآتى :

```
UPDATE ON <key field> FROM <alias>
  REPLACE <field1> WITH <exp1>
  [,<field2> WITH <exp2> ...][RANDOM]
```

حيث (key field) هو الحقل الفهرسي الذى يتم التحديث بناء عليه ويجب أن يكون الملفان مفهرسين (Indexed) ، أو مفروزين (Sorted) بناء على هذا الحقل وذلك لكى يقف المؤشر الخاص بكل ملف على نفس السجل فى الملفين مع كل حركة له . و(alias) هو الاسم المرادف الخاص بالملف الآخر المطلوب التحديث منه بالاضافة الى منطقة العمل الخاصة به . ويجب ملاحظة أن الملف المطلوب تعديله يكون مفتوحا وفى منطقة العمل التى تم اختيارها بواسطة الأمر (SELECT) .

والاختيار (REPLACE) يستخدم فى استبدال محتويات الحقول التى يراد تعديلها بمحتويات الحقول الموجودة فى الملف الآخر المستخدم فى التعديل . والاختيار (RANDOM) يستخدم عندما يكون ملف قاعدة البيانات المطلوب تعديله مفهرسا (Indexed) على الحقل الفهرسى (Key Field) وليس مفروزا (Sorted) .

مثـــال

عند تحديث بيانات الملف الرئيسى (Master) لقاعدة بيانات المخازن ، وذلك عن طريق ملف المبيعات (Sales) بناء على حقل رقم الجزء (Part_no) ، يتم كتابة الأوامر التالية :

```
SELECT 2
USE Sales
SELECT 1
USE MASTER INDEX MASTER
UPDATE ON Part_no FROM Sales ;
REPLACE Qty WITH Qty-Sales -> Qty
```

فى هذه الحالة يتم استبدال كمية المخزون (Qty) الموجودة فى الملف الرئيسى (Master) ، بنفس هذه القيمة (Qty) مطروحا منها قيمة المبيعات الموجودة فى حقل الكمية (Qty) الخاص بملف المبيعات (Sales) .

١٤٥ - الأمـــر (USE)

يستخدم هذا الأمر فى فتح ملف قاعدة البيانات وملفات الفهرس المرتبطة به ، واذا كان ملف قاعدة البيانات يحتوى على حقول ملاحظات (memo fields) يتم فتح ملف الملاحظات (.dbt) آليا . والصورة العامة لهذا الأمر كالآتى :

```
USE [<filename1>/?][INDEX <filename2>]
    [ALIAS <aliasname>]
```

والاسم (Filename1) هو اسم ملف قاعدة البيانات . ويمكن استخدام علامة الاستفهام (?) فى عرض ملفات قواعد البيانات المخزنة على القرص لاختيار الملف المطلوب منها .

والاسم (filename2) هو ملف الفهرس المرتبط بقاعدة البيانات المفتوحة . ويمكن فتح حتى سبعة ملفات فهرس مع ملف قاعدة بيانات واحد .

ويمكن استخدام الأمر (USE) دون كتابة أى شيء بعده ، وهذا يؤدى الى اغلاق جميع الملفات المفتوحة .

ويستخدم الاسم المرادف (alias name) فى تحديد اسم الملف متضمنا منطقة العمل المفتوحة .

١٤٦ - الأمر (WAIT)

يستخدم هذا الأمر في ايقاف تنفيذ البرنامج والانتظار حتى يضغط المستخدم على أى مفتاح . والصورة العامة له كالآتى :

```
WAIT  [<message>]  [TO <memvar>]
```

حيث (message) هى رسالة يتم عرضها للمستخدم لتنبيهه الى الضغط على أى مفتاح لاستمرار البرنامج .

والاختيار (TO <memvar>) يستخدم عندما يراد تخزين الحرف الذى يضغط عليه المستخدم فى متغير ذاكرة ، ويمكن كتابة الأمر (WAIT) دون كتابة أى شيء بعده ، وفى هذه الحالة يتم عرض الرسالة "Press any key to continue..". وهى الرسالة المبدئية (Default) .

مثال

لايقاف البرنامج مؤقتا واعطاء المستخدم الاختيار ليستمر أو ينهى تشغيل البرنامج ، يتم كتابة السطور التالية :

```
WAIT 'Do you want to continue? (Y/N)' TO mcon
IF UPPER (mcon) # 'Y'
     RETURN
ENDIF
```

١٤٧ - الأمـر (ZAP)

يستخدم هذا الأمر فى مسح جميع السجلات من ملف قاعدة البيانات المفتوح . والصورة العامة له كالآتى :

```
ZAP
```

وهذا الأمر يماثل استخدام الأمر (DELETE ALL) وبعده الأمر (PACK) . ويتبع ذلك مسح جميع ملفات الفهرس والملفات الأخرى المرتبطة بقاعدة البيانات .

الباب الثامن والعشرون

أهم الدوال المستخدمة

حتى يستطيع مخطط البرامج التحكم فى البرنامج وفى قاعدة البيانات ، يحتاج الى الالمام بمعظم الدوال (Functions) المستخدمة بواسطة برنامج (DBaseIII+) أو باقى برامج عائلة (DBase) مثل (DBaseIV) ، (FoxBase +) ، (FoxPro) . وفى هذا الباب يتم شرح معظم هذه الدوال مع توضيح وظائفها بالأمثلة كلما أمكن .

كما يستطيع القارىء الرجوع الى اللحق الموجود فى نهاية الكتاب لمعرفة الدوال الاضافية الخاصة ببرامج (DBaseIV) ، (FoxBase +) ، (FoxPro) .

ملاحظة

القيم الموجودة بين اقواس مربعة ([ ]) هى قيم اختيارية يستطيع المستخدم ادخالها او عدم ادخالها حسب الحاجة .

١ - الدالة (&)

تستخدم هذه الدالة فى التعويض بالماكرو (Macro Substitution) . ويحدث ذلك عندما يراد التعويض عن قيمة متغير معين فى مكان يعامل فيه هذا المتغير كحروف (Characters) . فمثلا عندما يراد البحث عن المتغير (mname) الذى يحتوى على الاسم (Mohamed) ، فان كتابة الأمر (FIND mname) لاتؤدى الى البحث عن الاسم (Mohamed) ولكنها تؤدى الى البحث عن الحروف (mname) . فى حين يمكن كتابة الأمر (FIND &mname) . وفى هذه الحالة يتم البحث عن محتويات المتغير (mname) وهى (Mohamed) . والصورة العامة لهذه الدالة كالآتى :

```
& <character variable> [.<exp>]
```

ويلاحظ أن المتغير فى هذه الحالة يجب أن يكون متغيرا حرفيا (Character Variable) . ويستخدم الاختيار (.<exp>) عندما يراد اضافة حروف معينة فى نهاية المتغير الحرفى . وفى هذه الحالة تستخدم النقطة (.) لتحديد نهاية المتغير الحرفى .

فمثلا لادخال علامة الضرب (X) داخل سلسلة حرفية مع اضافة أرقام بعد علامة الضرب يمكن كتابة السطور التالية :

```
STORE "X" TO alpha
STORE "15 & alpha.30"TO show
?   show
15 X 30
```

يلاحظ فى هذه الحالة ظهور العدد (15) ثم علامة الضرب (X) مكان المتغير (alpha) ثم العدد (30) كما كان مكتوبا فى السلسلة الحرفية .

وعندما يراد عرض رسالة معينة متضمنة اسم الشخص المطلوب عرض هذه الرسالة عليه ، يتم كتابة السطور التالية :

```
STORE "Mahmoud" TO mname
STORE "Hello & mname" TO greeting
?     greeting
```

يلاحظ فى هذه الحالة ظهور الرسالة "Hello Mahmoud".

واذا تغيرت محتويات المتغير (mname) الى أى اسم آخر ، فان الرسالة تتضمن الاسم الجديد مثل (Hello Magdy) مثلا .

واذا أراد مخطط البرامج اعطاء الفرصة للمستخدم لادخال اسم ملف قاعدة البيانات المطلوب استخدامه، فيمكنه مثلا انشاء متغير ذاكرة لاسم الملف مثل (dname) ، ثم يقوم المستخدم بادخال اسم الملف المطلوب فى هذا المتغير . وعن طريق دالة الماكرو (&) يمكن لمخطط البرامج فتح ملف قاعدة البيانات الذى أدخل المستخدم اسمه فيها وذلك كالآتى :

```
USE & dname
```

٢ - دالة القيمة المطلقة (ABS)

تستخدم هذه الدالة فى الحصول على القيمة المطلقة لقيمة عددية . والصورة العامة لها كالآتى :

```
ABS(<exp>)
```

حيث (exp) هى القيمة العددية المراد ايجاد القيمة المطلقة لها ، ويلاحظ أنها توضع بين قوسين .

وتستخدم هذه الدالة بصفة خاصة عندما يراد ايجاد الفرق العددى بين قيمتين دون الحاجة الى معرفة أيهما أكبر من الأخرى . كما يلاحظ من السطور التالية :

```
i = 20
J = 80
?    ABS(i-j)
```

وفى هذه الحالة يظهر الفرق (٦٠) موجبا .

٣ - الدالة (ASC)

تستخدم هذه الدالة فى الحصول على شفرة الآسكى الخاصة بأول حرف من سلسلة حرفية معينة . والصورة العامة لها كالآتى :

```
ASC (<exp>)
```

فمثلا عند كتابة السطر التالى :

```
ASC (Nagy)
```

يظهر العدد (78) الذى يمثل شفرة الآسكى الخاصة بالحرف (N) . وتستخدم هذه الدالة بصفة خاصة فى قوائم الاختيارات عندما يراد اختبار الحرف الذى يدخله المستخدم لتنفيذ أحد الاختيارات .

٤ - الدالة (AT)

تستخدم هذه الدالة فى البحث عن سلسلة حرفية فرعية (Substring) داخل سلسلة حرفية أخرى . وعندما تجدها فانها تعطى عددا يمثل ترتيب بداية هذه السلسلة الفرعية بالنسبة الى بداية السلسلة الحرفية الأخرى . واذا كانت السلسلة الحرفية التى يتم البحث عنها (Substring) غير موجودة داخل السلسلة الحرفية (String) ، فان هذه الدالة تعطى القيمة صفر (0) . والصورة العامة لهذه الدالة كالآتى :

```
AT(<exp1>, <exp2>)
```

حيث <exp1> هى السلسلة المطلوب البحث عنها داخل السلسلة الاكبر<exp2>

مثـال

يمكن كتابة السطر التالى :

```
?AT ("is" , "This is a test")
```

فى هذه الحالة يلاحظ ظهور الرقم (3). وذلك لأن الحروف (is) تبدأ من الحرف الثالث (بالرغم من تكرارها ابتداء من الحرف السادس ، لان الدالة (AT) تعطى رقم اول ظهور للسلسلة)

٥ - الدالـة (BOF)

تستخدم هذه الدالة لاختبار بداية الملف (Beginning of file). والصورة العامة لها كالآتى :

```
BOF()
```

وهى تعطى قيمة منطقية صحيح (True) أو غير صحيح (False). وتستخدم بصفة خاصة عندما يراد البحث عن سجل معين بفحص الملف عكسيا ، أى من نهاية الملف الى بدايته . فمثلا يمكن كتابة السطور التالية :

```
USE Client
GO BOTTOM
DO WHILE.NOT.BOF()
     IF Job = "Teacher"
          ? RECNO ()
     ENDIF
     SKIP -1
ENDDO
```

٦ - الدالة (CDOW)

تستخدم هذه الدالة للحصول على اسم اليوم فى تاريخ معين . والصورة العامة لها كالآتى :

```
CDOW(<exp>)
```

حيث (exp) قد يكون متغير ذاكرة تاريخى أو حقل تاريخى أو تاريخ اليوم الحالى .

فمثلا اذا كان تاريخ اليوم الحالى (DATE()) هو (02/17/90) فلايجاد اسم هذا اليوم يتم كتابة السطر التالى :

```
? CDOW (DATE())
```

وفى هذه الحالة يظهر الآتى :

```
Saturday
```

٧ - الدالة (CHR)

وتستخدم هذه الدالة في الحصول على الحروف والأعداد والحروف الخاصة عن طريق معرفة شفرة الآسكى الخاصة بهذه الحروف . والصورة العامة لها كالآتى :

```
CHR(<exp>)
```

حيث (exp) هو رقم صحيح من (١) الى (٢٥٥) .

وتستخدم هذه الدالة عندما يراد استخدام بعض الحروف الخاصة التى لايمكن كتابتها عن طريق لوحة المفاتيح في رسم أشكال معينة على الشاشة ، وكذلك فى تشغيل الجرس (Bell) لتحذير المستخدم عند حدوث خطأ معين .

فمثلا لتشغيل الجرس يتم كتابة السطر التالى :

```
? CHR(7)
```

ولاضافة رسالة تحذيرية بعد تشغيل الجرس يتم كتابة السطر التالى :

```
? CHR(7) + "Uncorrect....try again"
```

٨ - الدالـة (CMONTH)

تستخدم هذه الدالة في الحصول على اسم الشهر فى تاريخ معيـن . والصورة العامة لها كالآتى :

```
CMONTH(<exp>)
```

حيث (exp) هو متغير ذاكرة أو حقل تاريخى أو تاريخ اليوم الحالى .

مثـال

للحصول على اسم الشهر من تاريخ اليوم الحالى (02/17/90) يتم كتابة السطر التالى : .

```
? CMONTH (DATE())
```

وفى هذه الحالة يظهر اسم الشهر (February) .

ولتحديد اسم الشهر الذى يلى تاريخ اليوم الحالى بستين يوما مثلا ، يتم كتابة السطر التالى :

```
? CMONTH (DATE() +60)
```

وفى هذه الحالة يظهر اسم الشهر (April) .

٩ - الدالـة (COL)

تستخدم هذه الدالة في تحديد رقم العمود (Column) الذى يقف عنده المؤشر (Cursor) . ويحدث ذلك عندما يراد التحكم فى مكان المؤشر على الشاشة وعلى الطابعة من خلال البرنامج . والصورة العامة له كالآتى :

```
COL()
```

ويساعد ذلك على تحريك المؤشر الى أماكن مختلفة بالنسبة للمكان الحالى للمؤشر فمثلا يمكن كتابة السطر التالى :

```
@1,COL() + 5 SAY "Enter your name"
```

فى هذه الحالة يتم تحريك المؤشر خمسة أعمدة بعد آخر عمود كان يقف عنده .

١٠ - الدالــة (CTOD)

تستخدم هذه الدالة فى تحويل التاريخ الحرفى من حروف الى تاريخ . والصورة العامة لها كالآتى :

```
CTOD(<exp>)
```

وتستخدم عندما يراد مقارنة تاريخ بتاريخ آخر ، أو لتحديد الفترة الزمنية المحصورة بين تاريخ وتاريخ آخر .

مثــال

عندما يراد طباعة تقرير للبيانات الموجودة فى ملف قاعدة بيانات الطلبة ، وذلك بالنسبة للطلبة الملتحقين بالمعهد ابتداء من تاريخ معين يقوم المستخدم بادخاله وحتى تاريخ معين يقوم بادخاله أيضا ، فى هذه الحالة يتم كتابة السطور التالية :

```
STORE SPACE (8) TO start,end
@10,5 SAY " Enter start date " GET start
@12,5 SAY " Enter end date" GET end
READ
strat = CTOD (start)
end   = CTOD (end)
CLEAR
REPORT FORM Cadrep ;
FOR date_ent >= start .AND. date_ent <= end
```

ويلاحظ من السطور السابقة أنه تم تحويل التاريخ الذى يدخله المستخدم من حروف الى تاريخ ، وذلك لاستخدامه فى تحديد بداية ونهاية البيانات التى تظهر فى التقرير .

١١ - الدالــة (DATE)

تستخدم هذه الدالة فى الحصول على تاريخ اليوم الحالى . والصورة العامة لها كالآتى :

```
DATE()
```

مع ملاحظة أن التاريخ يظهر على الصورة الأمريكية وهي (mm/dd/yy) اذا لم يتم تغـــيير صورة التاريـخ بـــواسطـــــة الأمــــر (SET DATE) أو الأمـــــر (SET CENTURY) كما سبق الايضاح .

مثـــال

للحصول على تاريخ اليوم الحالى يتم كتابة السطر التالى :

```
? DATE()
```

وفى هذه الحالة يظهر التاريخ التالى مثلا : (02/20/90) .

ويمكن تخزين تاريخ اليوم الحالى فى متغير ذاكرة لاستخدامه بعد ذلك فى البرنامج وذلك كالآتى :

```
STORE DATE () TO mdate
```

١٢ - الدالـة (DAY)

تستخدم هذه الدالة فى الحصول على العدد الممثل لترتيب اليوم فى الشهر بالنسبة لتاريخ معين . والصورة العامة لها كالآتى :

```
DAY(<exp>)
```

حيث (exp) هو متغير ذاكرة تاريخى أو حقل تاريخى أو تاريخ اليوم الحالى مثلا .

مثـــال

للحصول على ترتيب اليوم الحالى وهو (02/17/90) بالنسبة للشهر ، يتم كتابة السطر التالى :

```
? DAY (DATE())
```

فى هذه الحالة يظهر الرقم (17) الذى يمثل ترتيب اليوم فى الشهر .

١٣ - الدالة (DBF)

تستخدم هذه الدالة فى الحصول على اسم ملف قاعدة البيانات المفتوح فى منطقة العمل التى تم اختيارها بواسطة الأمر (SELECT). والصورة العامة لهذه الدالة كالآتى :

```
DBF()
```

وفى حالة عدم وجود أى ملف قاعدة بيانات مفتوح ، تعطى هذه الدالة سلسلة حرفية خالية (Null String).

وتستخدم هذه الدالة عندما يراد معرفة اذا كان هناك ملف قاعدة بيانات (DBF) مفتوح قبل بداية البرنامج أم لا . وفى حالة وجود ملف مفتوح يتم اغلاقه ثم اعادة فتحه فى نهاية البرنامج حتى تعود حالة البرنامج الى وضعها الأصلى ، ولتنفيذ ذلك يتم كتابة السطور التالية :

```
Null = ""
IF   NULL < DBF()
     old_file =DBF()
     USE Clients
     -------
     -------
     -------   Commands
     -------
     -------
     USE & old_file
ENDIF
```

وفى هذا البرنامج يتم اختبار الشرط الموجود بعد (IF) للتأكد من وجود ملف قاعدة بيانات مفتوح قبل البرنامج . فاذا كان هناك ملف مفتوح ، يتم تخزين اسمه فى متغير الذاكرة (old_file) ، وذلك حتى يتسنى فتحه بعد ذلك ، ثم يتم اغلاقه عن طريق فتح ملف الموظفين (Clients) وتنفيذ الأوامر المطلوبة على هذا الملف . وقبل نهاية البرنامج يتم فتح الملف الذى كان مفتوحا مرة ثانية . ويلاحظ هنا استخدام دالة التعويض (&) لأن اسم الملف (DBF) ليس معروفا ولكنه مخزن فى متغير الذاكرة (old_file).

١٤ - الدالة (DELETED)

تستخدم هذه الدالة في تحديد السجلات التي تم وضع علامات عليها تمهيدا لمسحها . والصورة العامة لها كالآتي :

```
DELETED()
```

وهذه الدالة تعطي القيمة صحيح (True) اذا كان السجل الحالي تم وضع علامة عليه لمسحه (Marked for deletion)، ويفيد ذلك عندما يريد المستخدم استعراض السجلات التي تم وضع علامات عليها قبل مسحها نهائيا حتى يتأكد أنها السجلات المطلوب مسحها .

فمثلا لعرض بيانات السجلات التي تم تجهيزها للمسح يمكن كتابة السطر التالي :

```
DISPLAY FOR DELETED
```

١٥ - الدالة (DISKSPACE)

تستخدم هذه الدالة في تحديد حجم الذاكرة المتاح على القرص . والصورة العامة لها كالآتي :

```
DISKSPACE()
```

وهي تعطي عددا صحيحا يمثل عدد الحروف (Bytes) المتاحة على القرص . وتفيد عندما يراد عمل نسخ احتياطية من ملف قاعدة البيانات وكذلك عندما يراد عمل فرز للسجلات (Sorting)، حيث أن الفرز يتطلب انشاء ملف قاعدة بيانات جديد بالاضافة الى الملف الأصلي .

فمثلا اذا كان المتغير (mfilesize) يحتوي على عدد الحروف (Bytes) التي يتكون منها ملف قاعدة البيانات ، فيمكن كتابة السطور التالية لاجراء عملية الفرز (Sorting) :

```
IF DISKSPACE() > mfilesize * 2
     SORT ON Last_name TO Newfile
ELSE
     @ 20,15 SAY "There is not enough space"
ENDIF
```

١٦ - الدالة (DOW)

وتستخدم هذه الدالة فى الحصول على رقم يمثل ترتيب اليوم فى الأسبوع . والصورة العامة لها كالآتى :

```
DOW (<exp>)
```

حيث (exp) هو متغير ذاكرة تاريخى أو حقل تاريخى أو تاريخ اليوم الحالى ، مع ملاحظة أن الرقم (1) يمثل يوم الأحد (Sunday) . فمثلا للحصول على اليوم المثل للتاريخ الحالى يتم كتابة السطر التالى :

```
? DOW (DATE())
```

فى هذه الحالة يظهر الرقم (7) المثل ليوم السبت .

١٧ - الدالة (DTOC)

تستخدم هذه الدالة فى تحويل التاريخ الى حروف . والصورة العامة لها كالآتى :

```
DTOC (<EXP>)
```

حيث (exp) هو متغير ذاكرة تاريخى أو حقل تاريخى أو التاريخ الحالى . وتفيد هذه الدالة عندما يراد عرض الحقول التاريخية على الشاشة على صورة التاريخ المعروفة .

١٨ - الدالة (EOF)

تستخدم هذه الدالة فى تحديد نهاية ملف قاعدة البيانات (END of file) . والصورة العامة لها كالآتى :

```
EOF()
```

وهى تعطى القيمة المنطقية صحيح (True) عندما يصل مؤشر السجلات الى آخر الملف . وهذا لايعنى أن المؤشر يكون عند آخر سجل فى الملف ولكنه يتخطى هذا السجل ويصل الى علامة نهاية الملف .

وتستخدم هذه الدالة عندما يراد تنفيذ حلقة تكرارية على جميع سجلات قاعدة البيانات .

مثال

عندما يراد مثلا عرض بيانات الموظفين الذين تزيد أعمارهم عن (٤٠) سنة يتم كتابة الحلقة التكرارية التالية :

```
DO WHILE.NOT.EOF()
   LIST FOR age > 40
ENDDO
```

١٩ - الدالة (ERROR)

تستخدم هذه الدالة في الحصول على رقم يحدد الخطأ الذى قد يحدث أثناء تنفيذ البرنامج . والصورة العامة لها كالآتى :

```
ERROR()
```

وتستخدم بصفة خاصة عندما يراد علاج الأخطاء التى قد تحدث أثناء تنفيذ البرنامج ، وذلك عن طريق اغلاق بعض الملفات أو تغيير الأقراص أو مسح بعض الملفات لتوفير مساحة تخزينية . وهكذا .

مثال

يمكن ادخال السطر التالى فى برنامج لاكتشاف أى أخطاء قد تحدث به .

```
ON ERROR DO Err_prg WITH ERROR()
```

فعندما يحدث أى خطأ (Error()) يتم تنفيذ البرنامج الفرعى (Err_prg) مع ادخال الرقم الممثل لهذا الخطأ (ERROR) كمعامل (Parameter) للبرنامج الفرعى .

أما البرنامج الفرعى فى هذه الحالة فيكون كالآتى مثلا :

```
PARAMETERS Error_no
     IF Error_no =54
     -----
     -----  commands
     -----
     ENDIF
```

حيث الأوامر (Commands) فى هذه الحالة تكون رسائل للمستخدم لاتخاذ بعض الاجراءات للتخلص من هذا الخطأ .

٢٠ - الدالـة (FIELD)

تستخدم هذه الدالة للحصول على اسم أى حقل بمعلومية ترتيب هذا الحقل بين حقول قاعدة البيانات والصورة العامة لها كالآتى :

```
FIELD (<exp>)
```

حيث (exp) هو العدد الذى يمثل ترتيب الحقل ، وهو ينحصر بين (١) و (١٢٨) .

مثـال

للحصول على عدد حقول قاعدة البيانات من خلال البرنامج يتم كتابة السطور التالية :

```
USE Clients
num_fields = 0
null= ""
DO WHILE null < FIELD (num_fields + 1)
     num_fields = num_fields + 1
ENDDO
? num_fields
```

في هذا البرنامج يتم تنفيذ الحلقة التكرارية طالما كان اسم الحقل الذى يتم الحصول عليه بواسطة الدالة (FIELD (num_fields +1)) أكبر من السلسلة

الحرفية الخالية (null string)، وهذا يعنى أن الحلقة التكرارية تستمر طالما كان هناك حقول داخل الملف . وفى هذه الحالة يتم زيادة عدد الحقول واحدا . وهكذا يتم تحديد عدد الحقول بعد انتهاء تنفيذ الحلقة التكرارية .

٢١ - الدالة (FOUND)

هذه الدالة تعطى القيمة (صحيح) أى (True) عندما يصل البرنامج الى السجل الذى يتم البحث عنه بواسطة الأمر (FIND) أو الأمر (SEEK) أو الأمر (LOCATE) أو الأمر (CONTINUE) . والصورة العامة لها كالآتى :

```
FOUND()
```

وتستخدم هذه الدالة في البرنامج عندما يراد تنفيذ بعض الاجراءات فى حالة الوصول الى السجل المطلوب أو عدم الوصول اليه .

مثال

لاستخدام هذه الدالة مع الأمر (LOCATE) يمكن كتابة السطور التالية :

```
LOCATE FOR Job = "Teacher"
DO WHILE FOUND()
     ? name,address
     CONTINUE
ENDDO
```

وفى هذه الحالة يذهب المؤشر الى أول سجل يحقق الشرط ، فاذا وجد أول سجل، يتم تنفيذ الحلقة التكرارية التى يتم عن طريقها عرض بيانات حقول الاسم والعنوان الخاصة بهذا السجل ، ثم يتم البحث عن السجل التالى عن طريق الأمر (CONTINUE) . وهكذا يستمر تنفيذ الحلقة التكرارية طالما كان السجل موجودا فى كل مرة .

ويمكن استخدام هذه الدالة مع الأمر (SEEK) كالآتى مثلا :

```
SEEK "Teacher"
IF FOUND()
     DO WHILE Job = "Teacher"
       ? name, address
       SKIP
     ENDDO
ENDIF
```

ويراعى فى هذه الحالة أن يكون الملف مفهرسا على حقل الوظيفة (Job).

٢٢ - الدالة (IIF)

تستخدم هذه الدالة لادخال جملة (IF) الشرطية على سطر واحد بدلا من ادخالها على عدة سطور . والصورة العامة لها كالآتى :

```
IIF(<exp1>,<exp2>,<exp3>)
```

حيث (exp1) هو الشرط المراد اختباره ، فاذا تحقق فان الدالة تعطى القيمة (exp2) ، واذا لم يتحقق تعطى القيمة (exp3). وهى تسمى (IF) السريعة ، حيث أنها تؤدى الى سرعة تنفيذ البرنامج وزيادة كفاءته .

مثـال

يمكن ملاحظة الفرق بين استخدام الأمر (IF-ENDIF) واستخدام الدالة (IIF) من خلال هذا المثال ، حيث يتم كتابة السطور التالية التى توضح استخدام الأمر (IF-ENDIF).

```
IF sex = "F"
     mname = "Ms." + name
ESLE
     mname = "Mr." + name
ENDIF
```

وتنفيذ هذه السطور يؤدى الى تخزين الاسم الموجود فى حقل الاسم (name) فى متغير الذاكرة (mname)، مسبوقا بالحروف (.Ms) اذا كان السجل خاصا بأنثى (Female) ، أى أن السجل يحتوى على القيمة (F) فى حقل الجنس (Sex).

كما يخزنه مسبوقا بالحروف (.Mr) اذا كان السجل خاصا بذكر في الأحوال الأخرى ، أي السجلات التي تحتوى على أى قيمة أخرى غير (F) . واذا أريد استخدام الدالة (IIF) لتنفيذ نفس العملية ، يتم كتابة السطر التالى :

```
mname = IIF(sex = "F" , "Ms." , "Mr.") + name
```

وعند تنفيذ هذا السطر ، يتم اختبار الشرط (Sex = "F") فاذا تحقق يتم اضافة الحروف (.Ms) قبل الاسم الموجود فى الحقل (name) وتخزين القيمة الناتجة فى المتغير (mname) . واذا لم يتحقق ، يتم اضافة الحروف (.Mr) قبل الاسم الموجود فى الحقل name وتخزين القيمة الناتجة فى المتغير (mname) .

٢٣ - الدالة (INKEY)

هذه الدالة تعطى القيمة العددية المثلة لآخر حرف تم الضغط عليه بواسطة المستخدم . والصورة العامة لها كالآتى :

```
INKEY()
```

وهى تعطى عددا صحيحا بين (صفر) و (٢٥٥) يقابل شفرة الآسكى (ASCII Code) الخاصة بهذا الحرف ، وتستخدم عندما يراد اختبار الحروف التى يضغط عليها المستخدم .

مثال

السطور التالية توضح استخدام عداد للوقت يحدد الزمن الذى يقضيه المستخدم قبل ادخال الاختيار المطلوب .

```
DO WHILE.T.
     -----
     -----
     ----- Menu Options
     -----
     -----
     i = 0
     DO WHILE i = 0
          @ 1,72 SAY TIME()
```

```
        i = INKEY()
    ENDDO
    DO CASE
        CASE CHR(i)$ "Aa"
            DO <program1>
        CASE CHR(i)$ "Bb"
            DO <program2>
        CASE CHR(i)$ "Cc"
            DO <program3>
        CASE CHR(i)$ "Qq"
            RETURN
    ENDCASE
ENDDO
```

ويؤدى تنفيذ الحلقة التكرارية الداخلية الى عرض الوقت عن طريق الدالة (TIME()) طالما كانت (i=0) . حيث (i) تمثل القيمة العددية للحرف الذى يضغط عليه المستخدم ، لذلك يتم حساب الوقت المستهلك حتى يضغط المستخدم على أى حرف يمثل أحد الاختيارات الموجودة فى القائمة .

٢٤ - الدالة (INT)

تستخدم هذه الدالة فى تحويل القيم العددية الى أعداد صحيحة عن طريق حذف أى كسور عشرية . والصورة العامة لها كالآتى :

```
INT(<exp>)
```

مثال

لتحويل العدد (10.23) الى عدد صحيح يتم كتابة السطر التالى :

```
? INT(10.23)
```

ويلاحظ ظهور العدد (10) فى هذه الحالة .

٢٥ - الدالة (ISALPHA)

تستخدم هذه الدالة في اختبار أول حرف في قيمة معينة فاذا كان حرفا هجائيا فانها تعطى القيمة صحيح أى (TRUE) ، واذا كان رقما أو حرفا من الحروف الخاصة، فانها تعطى القيمة غير صحيح أى (False). والصورة العامة لهذه الدالة كالآتى :

```
ISALPHA(<exp>)
```

حيث (exp) هو القيمة التى يتم اختبارها .

مثـــال

يمكن كتابة السطر التالى :

```
? ISALPHA("abc123")
```

وفى هذه الحالة تظهر القيمة (.T.) أى صحيح . وذلك لأن أول حرف هو الحرف (a). كما يمكن كتابة السطر التالى :

```
? ISALPHA("123abc")
```

وفى هذه الحالة تظهر القيمة (.F.) أى غير صحيح . وذلك لأن أول حرف ليس حرفا هجائيا .

٢٦ - الدالـة (ISCOLOR)

تعطى هذه الدالة القيمة صحيح أى (True) اذا كان البرنامج يعمل على حالة الألوان (Color Mode). وتعطى القيمة غير صحيح أى (False) اذا كان يعمل على حالة اللون الأحادى (Monochrome). والصورة العامة لها كالآتى :

```
ISCOLOR()
```

وتفيد هذه الدالة في اعطاء مخطط البرامج امكانية التحكم في تصميم البرنامج بجعله يعمل على الشاشة الملونة أو الأحادية اللون حسب بيئة الحاسب المتوفرة .

مثـال

للتحكم فى الألوان من خلال البرنامج ، أى استخدام الألوان فى حالة الشاشة الملونة (Color Mode) ، واستخـــدام الأبيض والأسود فى حالة الشاشة الأحادية اللون ، يتم كتابة السطور التالية :

```
IF ISCOLOR()
     SET Color TO GR/B,W/R,GR
ELSE
     SET COLOR TO W+
ENDIF
```

٢٧ - الدالـة (ISLOWER)

تعطى هذه الدالة القيمة صحيح (True) عندما تبدأ القيمة الحرفية التى يتم اختبارها بحرف صغير (Lowercase) كما تعطى القيمة غير صحيح (False) عندما تبدأ بحرف كبير (Uppercase) . والصورة العامة لها كالآتى :

```
ISLOWER()
```

مثـال

لاختبار السلسلة الحرفية (abc123) يتم كتابة السطر التالى :

```
? ISLOWER ("abc123")
.T.
```

ويلاحظ هنا ظهور القيمة المنطقية (.T.) أى صحيح .

ولاختبار السلسلة الحرفية (ABC123) يتم كتابة السطر التالى :

```
? ISLOWER ("ABC123")
.F.
```

ويلاحظ هنا ظهور القيمة المنطقية (.F.) أى غير صحيح .

٢٨ - الدالة (ISUPPER)

وهي عكس الدالة السابقة . أي أنها تعطى القيمة صحيح أى (True) عندما تبدأ القيمة الحرفية التى يتم اختبارها بحرف كبير (Uppercase)، كما تعطى القيمة غير صحيح أو (False) عندما تبدأ بحرف صغير (Lowercase). والصورة العامة لها كالآتى :

```
ISUPPER ()
```

٢٩ - الدالة (LEFT)

هذه الدالة تعطى عددا من حروف السلسلة الحرفية بدءا من اليسار . والصورة العامة لها كالآتى :

```
LEFT (<exp1>,<exp2>)
```

حيث (exp1) هو السلسلة الحرفية المراد سحب جزء منها .

و (exp2) هو عدد يمثل عدد الحروف المطلوب استخراجها من اليسار .

وهذه الدالة تشبه الدالة (SUBSTR) مع اختلاف واحد وهو أنها لاتحتاج الى تحديد بداية السلسلة الحرفية المستخرجة ، حيث أنها تبدأ دائما من أول حرف من اليسار .

مثال

للحصول على الثلاثة حروف الأولى من الاسم (Mohamed) يتم كتابة السطر التالى .

```
? LEFT ("Mohamed",3)
Moh
```

يلاحظ ظهور الحروف (Moh) .

٣٠ - الدالــة (LEN)

هذه الدالة تعطى عددا يمثل عدد الحروف الموجودة فى سلسلة حرفية (String). والصورة العامة لها كالآتى :

```
LEN(<exp>)
```

حيث (<exp>) هو السلسلة الحرفية المراد حساب طولها .

مثــال

لايجاد طول الحقل (name) الذى يحتوى على الاسم (Hatem Zaky) يتم كتابة السطر التالى :

```
? LEN (name)
25
```

يلاحظ فى هذه الحالة ظهور الرقم (25) مع أن الاسم المذكور يحتوى على عشرة حروف فقط . وذلك لأن الدالة (LEN) تحسب طول الحقل بالكامل متضمنا الفراغات (Spaces).

ولحساب الطول الفعلى للاسم يتم التخلص من الفراغات (Spaces) الموجودة بعد الاسم باستخدام الدالة (TRIM) ، ثم حساب طول السلسلة الحرفية بعد ذلك . وذلك كالآتى :

```
? LEN (TRIM(name))
```

ويلاحظ فى هذه الحالة ظهور العدد (10) الممثل للعدد الفعلى للحروف متضمنا المسافة (Space) الموجودة بين الاسمين فقط ، حيث ان الدالة (TRIM) قد ازالت المسافات الموجودة آخر الاسم .

٣١ - الدالــة (LOG)

هذه الدالة تعطى قيمة اللوغاريتم الطبيعى لأى عدد .والصورة العامة لها كالآتى :

```
LOG (<exp>)
```

حيث (<exp>) هو العدد المطلوب ايجاد اللوغاريتم الطبيعى له . واللوغاريتم الطبيعى هـو الذى يكون أساسه النسبة التقريبية (e). فمثلا لايجاد اللوغاريتم للعدد (2.718) الذى يمثل النسبة التقريبية (e) يتم كتابة السطر التالى :

```
? LOG (2.718)
1.00
```

٣٢ - الدالة (LOWER)

تستخدم هذه الدالـة فى تحويل الحروف الكبيرة الى حروف صغيرة . والصورة العامة لها كالآتى :

```
LOWER(<exp>)
```

حيث (<exp>) هو السلسلة الحرفية المطلوب تحويلها الى حروف صغيرة ، وتستخدم هذه الدالة بصفة خاصة فى الحالات التى يراد فيها التحكم فى البيانات التى يدخلها المستخدم ، فمثلا اذا كان المطلوب من المستخدم ادخال بيانات الاسم (name) بحيث تكون بحروف صغيرة حتى تكون كل السجلات متماثلة ، فيتم استخدام هذه الدالـة فى تحويل البيانات التى يدخلها المستخدم الى حروف صغيرة (Lowercase) . فاذا أدخل المستخدم حروفا صغيرة أو كبيرة يتم تحويلها الى صغيرة .

مثال

عندما يراد البحث عن اسم معين فى قاعدة بيانات الطلبة مثلا يتم كتابة سطور التالية :

```
@ 15,15 SAY "Enter name to look for" ;
GET Lookup
READ
Lookup = LOWER (Lookup)
SEEK Lookup
```

يلاحظ فى هذه الحالة تحويل الاسم الذى يدخله المستخدم الى حروف صغيرة قبل

البحث عنه باستخدام الأمر (SEEK) . وذلك بفرض أن الأسماء قد سبق تخزينها فى قاعدة البيانات بحروف صغيرة .

٢٣ - الدالة (LTRIM)

تستخدم هذه الدالة فى مسح المسافات الخالية (Spaces) من أول السلسلة الحرفية من اليسار . والصورة العامة لها كالآتى :

```
LTRIM(<exp>)
```

وتفيد هذه الدالة عندما يراد علاج الأخطاء التى قد تنتج عن ادخال المستخدم لمسافات خالية قبل البيانات التى يقوم بادخالها ، حيث يتم مسح هذه المسافات قبل ادخالها الى الحقول .

فمثلا لكى يدخل المستخدم اسما معينا فى حقل الاسم (name) يتم أولا انشاء متغير ذاكرة لهذا الحقل مثل (mname) . ثم يتم استخدام الدالة السابقة فى التخلص من أى مسافات سواء فى أول الاسم أو فى آخره وذلك كالآتى مثلا :

```
mname = space(30)
@ 10,10 SAY "Enter a name" GET mname
READ
mname = LOWER (LTRIM(TRIM(mname)))
REPLACE name WITH mname
```

فى السطر الأول يتم انشاء متغير الذاكرة (mname) .

وفى السطر الثانى يتم عرض رسالة للمستخدم لادخال الاسم.

وفى السطر الثالث يتم تخزين الاسم فى المتغير (mname) .

وفى السطر الرابع يتم تحويل الاسم الى حروف صغيرة مع التخلص من المسافات فى أول الاسم وآخره باستخدام الدالتين (LTRIM) ، (TRIM) .

وفى السطر الخامس يتم استبدال محتويات حقل الاسم للسجل الحالى بالاسم الموجود فى متغير الذاكرة (mname) .

٣٤ - الدالة (LUPDATE)

هذه الدالة تعطى تاريخ آخر تحديث تم اجراؤه للملف . والصورة العامة لها كالآتى :

LUPDATE()

وتتيح هذه الدالة لمخطط البرامج التحكم فى تحديث المستخدم للبيانات حتى لايتم تحديثها عدة مرات ، حيث أن تحديثها عدة مرات قد يؤدى الى ادخال سجلات مكررة أو تجميع بيانات عددية أكثر من مرة مما يؤدى فى النهاية الى عدم دقة البيانات .

مثال

يمكن عن طريق السطور التالية عرض رسالة للمستخدم توضح له آخر تاريخ تم فيه تحديث الملف . ثم تترك له حرية الاختيار بين تحديث البيانات اذا كانت هناك بيانات جديدة مطلوب تحديثها بعد هذا التاريخ ، أو الاكتفاء بالتحديث الذى سبق اجراؤه .

```
IF LUPDATE() < DATE()
   entry = "?"
   @ 5,5 SAY "last entry was on"+DTOC(LUPDATE());+
         "Enter now ?(Y/N)" GET entry PICTURE "Y"
   READ
   IF entry = "y"
        DO <entry program>
   ENDIF
ENDIF
```

ويلاحظ في السطر الثالث استخدام علامة (+) ثم الفاصلة المنقوطة (;) وذلك لربط السلسلة الحرفية فى هذا السطر بالسلسلة الحرفية المكملة لها فى السطر التالى . وعند الرغبة فى تحديث بيانات الملف يتم ادخال (Y) فيتم تنفيذ البرنامج (entry program) الذى يؤدى الى ادخال البيانات المطلوب تحديثها .

٣٥ - الدالة (MAX)

هذه الدالة تعطى أكبر قيمة من قيمتين عدديتين .والصورة العامة لها كالآتى :

```
MAX(&lt;exp1&gt;, &lt;exp2&gt;)
```

٣٦ - الدالة (MIN)

هذه الدالة عكس الدالة السابقة ، حيث تعطى أقل قيمة من قيمتين عدديتين . والصورة العامة لها كالآتى :

```
MIN(<exp1>,<exp2>)
```

وتستخدم هذه الدالة عندما يراد مثلا الحصول على أقل قيمة عددية لحقل معين .

فمثلا للحصول على أقل مرتب لموظف فى قاعدة بيانات الموظفين (Clients) يتم كتابة السطور التالية :

```
USE Clients
min_sal = salary
DO WHILE.NOT.EOF()
    min_sal = MIN(min_sal,salary)
    SKIP
ENDDO
USE
RETURN
```

فى هذا البرنامج يتم الانتقال الى السجل التالى دائما بواسطة الأمر (SKIP) ، ثم يتم الحصول على أقل قيمة من المرتب (Salary) والمرتب السابق ثم يتم تخزين هذه القيمة فى المتغير (min_sal)، وهكذا تتكرر هذه العملية حتى يتم الحصول على أقل قيمة للمرتب .

ملاحظة

يمكن استخدام نفس الطريقة فى الحصول على أكبر قيمة بواسطة الدالة (MAX).

٣٧ - الدالة (MOD)

هذه الدالة تعطى باقى القسمة الصحيحة لعدد على عدد آخر . والصورة العامة لها كالآتى :

```
MOD(<exp1>, <exp2>)
```

وتستخدم هذه الدالة بصفة خاصة فى التحويل من وحدات الى أخرى ، مثل تحويل الياردة الى بوصة والمتر الى سنتيمتر والساعة الى دقائق وثوان . وهكذا .

فمثلا اذا كان هناك عدد من الدقائق يراد تحويله الى عدد من الأيام وعدد من الساعات وعدد من الدقائق ، يمكن كتابة السطور التالية :

```
t = 36500
minutes = MOD(t,60)
h = INT(t/60)
hours = MOD(h,24)
days = INT(h/24)
? t,"minutes are:",days,"days",hours,"hours";
  minutes, "minutes"
```

وعند الضغط على مفتاح الادخال بعد السطر الأخير يلاحظ ظهور الآتى :

```
36500 minutes are:25 days 8 hours 20 minutes
```

والسطر الأول من البرنامج يتم عن طريقه انشاء المتغير (t) الذى يحتوى على العدد (36500) دقيقة .

والسطر الثانى يقوم بتحديد عدد الدقائق التى تبقى بعد القسمة على (٦٠) . وهى الدقائق التى تبقى بعد تحديد الأيام والساعات .

والسطر الثالث يقوم بتحديد عدد الساعات الصحيحة الموجودة الناتجة عن قسمة عدد الدقائق على (٦٠) .

والسطر الرابع يقوم بتحديد الساعات الباقية بعد القسمة على (٢٤) .

والســـطر الخامس يحدد عدد الأيام الصحيحة الناتجة عن قسمة الساعات على (٢٤) .

والسطر السادس يقوم بعرض البيانات التى تم حسابها على الشاشة .

٣٨ - الدالــة (MONTH)

هذه الدالة تعطى عددا يمثل ترتيب الشهر فى السنة بالنسبة لتاريخ معين . والصورة العامة لها كالآتى :

```
MONTH(<exp>)
```

حيث (exp) هو متغير ذاكرة حرفى أو حقلا حرفيا أو تاريخ اليـــوم الحالى .

فمثلا اذا كان تاريخ اليوم الحالى (2/18/1990) ، يمكن الحصول على رقم الشهر كالآتى :

```
? MONTH(DATE())
```

فى هذه الحالة يظهر الرقم (2) الممثل لترتيب الشهر .

٣٩ - الدالــة (NDX)

هذه الدالة تعطى اسم ملف الفهرس المفتوح فى منطقة العمل (Work Area) التى سبق اختيارها بواسطة الأمر (SELECT) . والصورة العامة لهذه الدالة كالآتى :

```
NDX(<exp>)
```

حيث (exp) هو رقم يمثل ترتيب ملف الفهرس بين الملفات المفتوحة ، وهو يأخذ أى رقم من (1) الى (7) .

مثـــال

لكى يعرف البرنامج أسماء ملفات الفهرس المفتوحة ويتعامل معها يمكن كتابة السطور التالية :

```
i =1
Null= ""
DO WHILE Null < NDX(i).AND.i <= 7
     ? NDX(i)
     i =i +1
ENDDO
```

٤٠ - الدالـة (OS)

هـذه الدالة تعطى اسم نظام التشغيل الذى يعمل عليه البرنـامـج . والصـورة العامة لها كالآتى :

```
OS()
```

وتستخدم هذه الدالة عندما يراد تصميم برنامج نقال (Portable) ، أى يمكنه العمل على نظم تشغيل مختلفة مثل (MS-DOS) ، (UNIX) .

مثـــال

لتحويل البرنامج الذي يعمل على نظام التشغيل (MS-DOS) ليعمل على نظام التشغيل (UNIX) ، يتم كتابة السطور التالية :

```
STORE OS() TO opsys
IF SUBSTR(opsys,1,4) = "UNIX"
     DO setunix
ENDIF
```

وتـؤدى هـذه السطـور الى اختبار نظام التشغيل المستخـدم فـاذا كـان الـنظام (UNIX) يتم تنفيذ البرنامج (setunix) الذى يؤدى الى تجهيز البرنامج للتعامل مع نظام التشغيل (UNIX) .

٤١ - الدالة (PCOL)

تستخدم هذه الدالة فى تحديد العمود (Column) الذى يقف عنده رأس الطباعة بالنسبة للورقة الموجودة على الطابعة . والصورة العامة لها كالآتى :

```
PCOL()
```

ويمكن عن طريق هذه الدالة تحريك رأس الطباعة على الورقة فى أعمدة مختلفة بالنسبة للعمود الذى يقف عنده .

فمثلا عندما يراد الطباعة بعد آخر طباعة سبق تنفيذها بخمسة أعمدة يتم كتابة السطور التالية :

```
SET DEVICE TO PRINT
@ 1,PCOL()+ 5 SAY "This is a test"
SET DEVICE TO SCREEN
```

كما يمكن عن طريق هذه الدالة معرفة رقم العمود الذى يقف عنده رأس الطباعة بالنسبة للورقة . كالآتى مثلا :

```
? PCOL()
```

فى هذه الحالة يظهر العدد (5) مثلا . أى أن رأس الطباعة يقف عند العمود (5) من الورقة ، وهكذا .

٤٢ - الدالة (PROW)

تستخدم هذه الدالة فى تحديد السطر (Row) الذى يقف عنده رأس الطباعة بالنسبة للورقة . والصورة العامة لها كالآتى :

```
PROW()
```

كما تستخدم أيضا فى تحريك رأس الطباعة عددا من السطور بالنسبة لآخر سطر كان يقف عنده .

فمثلا عندما يراد الطباعة بعد آخر طباعة سبق تنفيذها بخمسة سطور (PROW()+5) ، يتم كتابة السطور التالية :

```
SET DEVICE TO PRINT
@ PROW()+ 5,1 SAY "This is a test"
SET DEVICE TO SCREEN
```

٤٣ - الدالة (RECCOUNT)

هذه الدالة تعطى عدد السجلات (Records) فى ملف قاعدة البيانات المفتوح . والصورة العامة لها كالآتى :

```
RECCOUNT()
```

وتستخدم هذه الدالة بصفة خاصة في البرامج التى يتم من خلالها عمل نسخ احتياطية (Backups) لملفات قواعد البيانات آليا . حيث يتم استخدامها مع الدالة (RECSIZE()) والدالة (DISKSPACE()) فى تحديد الحجم المتاح على القرص واذا كان يكفى لعمل نسخة احتياطية للملف أم لا .

مثال

لايجاد عدد السجلات فى ملف بيانات الطلبة يتم كتابة السطور التالية :

```
USE cadets
?   RECCOUNT
```

فى هذه الحالة يظهر العدد الذى يمثل عدد السجلات فى الملف .

٤٤ - الدالة (RECNO)

تستخدم هذه الدالة فى تحديد رقم السجل (Record number) الذى يقف عنده المؤشر . والصورة العامة لها كالآتى :

```
RECNO()
```

وتفيد هذه الدالة بصفة خاصة بعد أوامر البحث مثل الأمر (SEEK) لمعرفة اذا كان هناك سجل يحقق الشرط أم لا .

فمثلا عندما يراد البحث عن اسم معين موجود فى المتغير (Lookup) يتم كتابة السطور التالية :

```
SEEK Lookup
Recno = RECNO()
IF Recno>0
     SET FORMAT TO Cadets
     READ
     CLOSE FORMAT
ELSE
     @ 10,10 SAY "There is no & lookup"
     ? CHR(7)
ENDIF
```

وفى هذا البرنامج يتم البحث عن الاسم الموجود فى متغير الذاكرة (Lookup) فاذا كان موجودا يتم تخزين رقم هذا السجل في متغير الذاكرة (Recno) . واذا لم يكن موجودا يتم تخزين القيمة صفر فى هذ المتغير . وفى الحالة الأولى يتم فتح شاشة الادخال عن طريق الأمر (SET FORMAT) ليقوم المستخدم بادخال البيانات المطلوبة . وفي الحالة الثانية يتم عرض الرسالة المبينة . ويلاحظ أهمية استخدام الماكرو فى هذه الرسالة لاظهار محتويات المتغير (Lookup) للمستخدم ليعرف أن هذا الاسم غير موجود .

٤٥ - الدالـة (RECSIZE)

هذه الدالة تعطى حجم السجل (Record Size) فى ملف قاعدة البيانات المفتوح . والصورة العامة لها كالآتى :

```
RECSIZE()
```

وتستخدم هذه الدالة بصفة خاصة في البرامج التطبيقية التي يتم من خلالها عمل نسخ احتياطية (Backups) للفات قواعد البيانات المستخدمة ، وذلك بالاشتراك مع الدالة (RECCOUNT()) والدالة (DISKSPACE()) . حيث تساعد على التأكد من

وجود مساحة متاحة فى القرص لتخزين النسخة الاحتياطية .

فمثلا لعمل نسخة من ملف قاعدة بيانات كبير يحتاج الى عدة أقراص لتخزينه، يتم كتابة السطور التالية :

```
USE File1
SET DEFAULT TO B
DO WHILE .NOT. EOF()
   WAIT "Insert new disk in drive B, and press a key."
   COPY NEXT(DISKSPACE() - <header size>)/RECSIZE() ;
   TO Backup
SKIP
ENDDO
USE
```

حيث (header Size) هو حجم العنوان الذى يكون موجودا قبل كل سجل ويتم حسابه من العلاقة الآتية :

```
header size =32*<number of fields> + 35
```

والبرنامج السابق يؤدى الى استمرار نسخ السجلات سجلا سجلا طالما كانت القيمة الناتجة بعد الأمر (COPY NEXT) أكبر من واحد . وعندما تقل هذه القيمة عن واحد ، فان هذا يعنى أن المساحة الخالية (Disk Space) المتاحة على القرص أقل من حجم السجل التالى ، وبالتالى يتوقف النسخ ويطلب البرنامج من المستخدم وضع قرص جديد . ثم تتكرر هذه العملية حتى يتم نسخ ملف قاعدة البيانات .

٤٦ - الدالة (REPLICATE)

تستخدم هذه الدالة فى تكرار حرف معين أو قيمة حرفية معينة عددا من المرات يتم تحديده . والصورة العامة لهذه الدالة كالآتى :

```
REPLICATE(<exp1>,<exp2>)
```

حيث (exp1) هو القيمة الحرفية المراد تكرارها .

و (exp2) هو العدد الذى يمثل عدد مرات تكرار هذه القيمة .

ويجب ملاحظة أن عدد الحروف الذى يتكون من عملية التكرار يجب ألا يزيد عن (٢٥٤) حرفا .

وتستخدم هذه الدالة بصفة خاصة في تكوين أشكال على الشاشة مثل المستطيلات التى يتم تكوين أضلاعها من حروف معينة مثل الحرف (*) أو عن طريق استخدام أى حروف أخرى يتم تكوينها باستخدام الدالة (CHR()) .

فمثلا للحصول على خط أفقى مكون من تكرار الحرف (*) عددا محددا من المرات يتم كتابة السطر التالى :

```
5,5 SAY REPLICATE("*",20)
```

وفى هذه الحالة يظهر الآتى على الشاشة :

```
* * * * * * * * * * * * * * * * * * * *
```

٤٧ - الدالة (RIGHT)

هذه الدالة تعطى عددا من الحروف الموجودة فى سلسلة حرفية بدءا من اليمين والصورة العامة لها كالآتى :

```
RIGHT(<exp1> , <exp2>)
```

حيث (exp1) هو القيمة الحرفية المراد استخراج عدد من حروفها .
و (exp2) هو العدد الذى يمثل عدد الحروف المراد استخراجه . وتستخدم هذه الدالة فى حالات كثيرة يراد فيها استخراج جزء من سلسلة حرفية معينة

مثال

لاستخراج الحروف الثلاثة الأخيرة من الاسم (Mahmoud) يتم كتابة السطر التالى :

```
? RIGHT("Mahmoud",3)
```

فى هذه الحالة يلاحظ ظهور الحروف الثلاثة (oud) .

٤٨ - الدالـة (ROUND)

هذه الدالة تقوم بتقريب العدد لعدد محدد من الكسور العشرية . والصورة العامة لها كالآتى :

```
ROUND(<exp1> , <exp2>)
```

حيث (exp1) هو العدد المطلوب تقريبه .
و (exp2) هو عدد الكسور العشرية المطلوب التقريب اليها .

مثـال

لتقريب العدد (10.765788) لأقرب رقمين عشريين يتم كتابة السطر الآتى :

```
? ROUND(10.765788 ,2)
```

وعند الضغط على مفتاح الادخال يظهر العدد (10.77) .

٤٩ - الدالـة (ROW)

هذه الدالة تعطى السطر الحالى الذى يقف عنده المؤشر على الشاشة . والصورة العامه لها كالآتى :

```
ROW()
```

وتستخدم عندما يراد التحكم فى مكان المؤشر وتحريكه عددا من السطور بالنسبة للمكان الذى يقف عنده . فمثلا يمكن كتابة السطر التالى :

```
@ ROW()+5,3 SAY "Enter your name"
```

فى هذه الحالة تظهر الرسالة المبينة بعد خمسة سطور من آخر سطر وصل اليه المؤشر .

٥٠ - الدالـة (RTRIM)

تستخدم هذه الدالة فى مسح المسافات من نهاية قيمة حرفية معينة . والصـورة العامه لها كالآتى :

```
RTRIM(<exp>)
```

وهى تماثل الداله (TRIM()) تماما .

٥١ - الدالـة (SPACE)

تستخدم هذه الدالة فى انشاء متغير ذاكرة يحتوى على عدد معين من الحـروف الخالية (blanks) . والصورة العامة لها كالآتى :

```
SPACE(<exp>)
```

ويمكن تكوين متغير ذاكرة يحتوى على عدد من الحروف يصل الى (٢٥٤) حرفا .

مثـال

عندما يراد مثلا انشاء متغير ذاكرة (mname) حتى يستطيع المستخدم ادخـال الاسم فيه يتم كتابة السطور التالية :

```
mname = SPACE(30)
@5,5 SAY "Enter new name" GET mname
READ
```

٥٢ - الدالـة (SQRT)

هذه الدالة تعطى الجذر التربيعى للقيمة العددية الموجبة . والصورة العامة لها كالآتى :

```
SQRT(<exp>)
```

مثـــال

لايجاد الجذر التربيعى للعدد (4) يتم كتابة السطر التالى :

```
? SQRT(4)
```

فى هذه الحالة يظهر العدد (2.00).

٥٣ - الدالـة (STR)

تستخدم هذه الدالة فى تحويل القيمة العددية الى سلسلة حرفيـة (String). والصورة العامة لها كالآتى :

```
STR(<exp>, <length> , <decimal>)
```

حيث (exp) هو القيمة العددية المراد تحويلها .

و (length) هو عدد الأرقام المراد ظهورها وهو اختيارى ، وفى حالة عدم ادخاله يظهر حتى عشرة أرقام .

و (decimal) هو عدد الأرقام العشرية وهو اختيارى أيضا ، وفى حالة عدم ادخاله يتم التقريب لأقرب عدد صحيح .

وعند ادخال طول (length) أصغر من عدد الأرقام الصحيحة الموجودة فى العدد ، فان البرنامج يعرض مجموعة من حروف النجمة (*) مكان العدد . وعند ادخال عدد أرقام عشرية (Decimal) أقل من عدد الأرقام العشرية الموجود فى العدد ، يتم تقريب الأرقام العشرية الزائدة .

مثـــال

لعرض العدد (33.56) كحروف (String) ، يتم كتابة السطر التالى :

```
? STR(33.56 ,4,1)
```

فى هذه الحالة يظهر العدد (33.6).

ويلاحظ هنا عند تحديد الطول (Length) تم حساب نقطة الكسر العشرى (Decimal Point) ضمن عدد الأرقام ، فأصبح العدد (4) بدلا من (3) .

٥٤ - الدالـة (STUFF)

تستخدم هذه الدالة فى تغيير أى جزء داخل سلسلة حرفية معينة . والصورة العامة لها كالآتى :

```
STUFF(<exp1>,<start position>,<number of characters>;
,<exp2>)
```

حيث

exp1	هى السلسلة الحرفية المطلوب التعديل فيها .
و exp2	هى السلســـلة الحرفية المطلوب ادخالها فى السلسلة الحرفية الأولى .
و start position	هى قيمة عددية تمثل المكان المطلوب ادخال السلسلة الحرفية بدءا منه .
number of characters	هو عدد الحروف المطلوب استبدالها من السلسلة الحرفية الأولى . واذا كان هذا العدد صفرا ، يتم ادخال السلسلة الحرفية الثانية داخل السلسلة الحرفية الأولى دون تغيير الحروف الموجودة فى السلسلة الأولى . أى يتم حشر السلسلة الثانية داخل السلسلة الأولى (Insertion) . واذا كانت السلسلة الثانية عبارة عن سلســـلة خاليـــة (null string) ، يتم مسح حروف من السلسلة الأولى بقدر طول السلسلة الخالية .

مثـال

اذا أريد تغيير عنوان معين داخل قاعدة بيانات الطلبة (Cadets) تكتب السطور التالية :

```
. new_street = nasr city
. USE Cadets
. GO 5
. ? address
10 -Ainshams -Cairo
. REPLACE address WITH STUFF(address,4,9,new_street)
. ?address
```

يلاحظ فى هذه الحالة ظهور العنوان (10-nasr city-Cairo) بدلا من العنوان السابق .

٥٥ - الدالة (SUBSTR)

تستخدم هذه الدالة فى استخراج جزء من سلسلة حرفية معينة . والصورة العامة لها كالآتى :

```
SUBSTR(<exp1>,<starting position>
       ,[<number of characters>])
```

حيث (exp1) هو السلسلة الحرفية المطلوب استخراج جزء منها .

و (starting position) هو المكان الذى يبدأ منه استخراج السلسلة الحرفية الفرعية .

و (number of characters) هو عدد اختيارى يمثل عدد الحروف المراد سحبها من السلسلة الحرفية . وفي حالة عدم كتابة هذا العدد يتم استخراج السلسلة الحرفية بدءا من مكان البداية (starting position) الى آخر السلسلة الحرفية .

مثال

فى المثال السابق الخاص بالعنوان الموجود فى السجل الخامس . يراد معرفة المدينة التى يقع فيها هذا العنوان . لتنفيذ ذلك يتم كتابة السطر التالى :

```
?SUBSTR(address,14,5)
```

فى هذه الحالة يظهر الآتى :

```
Cairo
```

٥٦ - الدالـة (TIME)

هذه الدالة تعطى الوقت الحالى الذي تم ادخاله عند بدء تشغيل الجهاز من خلال نظام التشغيل . والصورة العامة لها كالآتى :

```
TIME()
```

مثـال

للحصول على الوقت الحالى يتم كتابة السطر التالى :

```
?TIME()
```

وعند الضغط على مفتاح الادخال يظهر الآتى مثلا :

```
20:45:20
```

٥٧ - الدالـة (TRANSFORM)

تستخدم هذه الدالة للتحكم فى شكل المخرجات التى تنتج من الأوامر (?,??,DISPLAY,LABEL,LIST,REPORT) . والصورة العامة لها كالآتى :

```
TRANSFORM(<exp1> , <exp2>)
```

وهى تؤدى نفس العمل الذى تؤديه عبارة (PICTURE) مع الأمر (@...SAY).

مثـال

لعرض الأسماء الموجودة فى الحقل (name) بحيث يتم فصل كل حرف عن الحرف التالى بمسافة (Space) ، يتم كتابة السطر التالى :

```
DISPLAY TRANSFORM(name,@Rxxxxxxxx)
```

فاذا كان الاسم الموجود فى هذا الحقل هو (Mohamed) مثلا يظهر الآتى :

```
M o h a m e d
```

ولعرض مرتب أربعة موظفين بحيث يتكون المرتب من ثمانية أرقام مع رقمين عشريين يتم كتابة السطر التالى :

```
LIST NEXT 4 TRANSFORM(salary, "# # # #. # #")
```

فى هذه الحالة تظهر الأعداد كالآتى :

Record #	Salary
1	570.50
2	600.80
3	1000.00
4	700.00

٥٨ – الدالة (TRIM)

تستخدم هذه الدالة فى مسح المسافات الخالية من نهاية السلسلة الحرفية والصورة العامه لها كالآتى :

```
TRIM(<exp>)
```

وهى تفيد عندما يراد التأكد من التخزين الصحيح للبيانات التى يدخلها المستخدم ، حيث يتم أولا ادخال هذه البيانات فى متغير ذاكرة بعد التخلص من المسافات الخالية فى أوله أو فى آخره ، ثم يتم نقل هذه البيانات من متغير الذاكرة الى الحقل الخاص بها .

فمثلا عندما يراد ادخال أى اسم فى الحقل (name) ، يتم كتابة الأمر التالى لانشاء متغير الذاكرة (mname)

```
STORE SPACE(30) TO mname
```

ثم يتم التخلص من المسافات الخالية فى أول الاسم وآخره كالآتى :

```
STORE LTRIM(TRIM(mname)) TO mname
```

وهذا يؤدى الى ادخال الحروف التى يدخلها المستخدم دون أى مسافات قبلها أو بعدها .

وتستخدم هذه الدالة أيضا عندما يراد معرفة الطول الصحيح لأى سلسلة حرفية باستخدام الدالة (LEN) وذلك كالآتى مثلا :

```
LEN(LTRIM(TRIM(mname)))
```

٥٩ - الدالـة (TYPE)

تستخدم هذه الدالة فى تحديد نوع أى قيمة أو متغير معين . والصورة العامة لها كالآتى :

```
TYPE(<exp>)
```

وهى تعطى حرف كبير (Capital) يمثل نوع القيمة (exp)، مثل (C) للقيم الحرفية (Characters) ، (N) للقيم العددية (Numeric) ، (L) للقيم المنطقية (Logical) ، (M) لحقول الملاحظات (Memo) ، (U) للقيم غير المعرفة (Undefined).

مثـال

عندما يراد اختبار المتغير (score) مثلا يتم كتابة السطر التالى :

```
? TYPE("score")
```

فى هذه الحالة يظهر الحرف (U) ، وهذا يعنى أن المتغير غير معرف (Undefined) . وذلك لأن المتغير لم يتم تعريفه قبل هذا الأمر . أما عند كتابة الآتى مثلا :

```
STORE 100 TO score
TYPE("score")
```

فى هذه الحالة يظهر الحرف (N) ، وهذا يعنى أن المتغير عددى .

٦٠ - الدالة (UPPER)

تستخدم هذه الدالة في تحويل الحروف الصغيرة (Lowercase) الى حروف كبيرة (Upercase). والصورة العامة لها كالآتى :

```
UPPER(<exp>)
```

وتفيد هذه الدالة فى التأكد من ادخال البيانات التى يدخلها المستخدم بنفس شكل البيانات المخزنة فى الملف ، كما تفيد أيضا عندما يتم عرض رسالة على المستخدم واستقبال الرد على هذه الرسالة والتعامل مع هذا الرد بصرف النظر عن ادخاله بحروف كبيرة أو صغيرة .

مثـــال

عندما يراد البحث عن اسم معين فى حقل الاسم (name) ، يتم أولا انشاء متغير ذاكرة (Lookup) مثلا لاستقبال الاسم الذى يدخله المستخدم، ثم تحويل هذا المتغير الى حروف كبيرة حتى يماثل الحروف الموجودة فى حقل الاسم لجميع السجلات ، ثم يتم البحث عن هذا الاسم باستخدام الأمر (SEEK). ولتنفيذ ذلك يتم كتابة السطور التالية :

```
Lookup = SPACE(15)
@10,10 SAY "Enter name of person to edit" GET Lookup
READ
Lookup = UPPER(Lookup)
SEEK Lookup
```

٦١ - الدالة (VAL)

تستخدم هذه الدالة في تحويل الأرقام الموجودة فى السلسلة الحرفية (String) الى العدد المقابل . والصورة العامة لها كالآتى :

```
VAL(<exp>)
```

وهى تقوم بعكس العمل الذى تؤديه الدالة (STR)، مع ملاحظة أن البيانات الموجودة فى القيمة (exp) يجب أن تكون أعدادا وليست حروفا .

مثـــال

يمكن كتابة السطور التالية :

```
STORE "88.50" TO string
VAL(string)
```

فى هذه الحالة يظهر نفس العدد (88.50) ولكن الفرق هنا أن هذه القيمة عددية ، أى يمكن التعامل معها بالجمع عليها أو الطرح منها، وهكذا .

٦٢ - الدالـة (VERSION)

هذه الدالة تعطى رقم نسخة برنامج (DBaseIII+) أو أى برنامج آخر من برامج عائلة (DBase) مثل (DBaseIV) ، (FoxBase +) ، (FoxPro) المستخدمة . والصورة العامه لها كالآتى :

```
VERSION()
```

وتستخدم فى البرامج التى تتطلب بعض الخصائص المرتبطة بنسخة البرنامج المستخدمة .

٦٣ - الدالـة (YEAR)

تستخدم هذه الدالة فى الحصول على العدد الممثل للسنة داخل تاريخ معين . والصورة العامة لها كالآتى :

```
YEAR(<exp>)
```

حيث (exp) هو متغير ذاكرة تاريخي أو حقل تاريخي أو تاريخ اليوم الحالي .

مثال

اذا كان تاريخ اليوم الحالي هو (02/18/90) ، فيمكن الحصول على العدد الممثل للسنة كالآتي :

```
?YEAR(DATE())
```

يلاحظ في هذه الحالة ظهور العدد (1990) .

ملحـــق (١)

أهم الأوامر والدوال المستخدمة

فى برنامج (DBase IV)

هذا الملحق يضم الأوامر والدوال الأضافية فقط ، حيث أن باقى الأوامر والدوال مطابقة تماما لما سبق عرضه فى برنامج (+ DBase III) .

أولا : الأوامـــر (Commands)

١ - الأمر (???)

وهو يؤدى الى ارسال المخرجات مباشرة الى الطابعة .

٢ - الأمر (@... SAY... GET)

وهو نفس الأمر الخاص ببرنامج (+DBase III) مع اضافة امكانيات أخرى له مثل ادخال شروط (Conditions) لظهور بيانات الـ (Gets) وظهور رسائل الأخطاء والتحكم في ألوان الأعمدة الضوئية . ولمزيد من الايضاح ارجع الى الباب الخاص بالأوامر فى لكتاب الأول.والصورة العامة له كالآتى :

```
@   <ROW> , <COL>
    [SAY  <expression> [PICTURE <expC>]
       [FUNCTION <function list>]]
    [GET <variable>
       [[OPEN WINDOW <window name>]
       [PICTURE <expC>]
       [FUNCTION <function list>]
       [RANGE [<low>][,<high>]]
       [VALID <condition> [ERROR <expC>]]
       [WHEN <condition> [DEFAULT <expression>]
       [MESSAGE <expC>]]
       [COLOR [<stand>][,<enhanced>]]
```

٣ - الأمـر (@... FILL)

وهو يؤدى الى رسم مستطيل مظلل بلون محدد

```
@  <row1> , <col1>  FILL TO <row2> , <col2>
        [COLOR <color attribute>]
```

٤ - الأمر (@... TO)

وهو نفس الأمر المستخدم في برنامج (+DBase III) لرسم مستطيل على الشاشة مع بعض الاضافات للتحكم في لون الخطوط .ارجع الى الباب الخاص بالأوامر فى الكتاب الاول.والصورة العامة كالآتى :

```
@ <row1> , <col1> TO <row2> ,<col2>
  [ DOUBLE / PANEL / <border definition string>]
  [ COLOR <color attribute> ]
```

٥ - الأمر (ACTIVATE MENU)

ويستخدم لتشغيل عمود قوائم (Menu Bar) سبق تصميمه .

٦ - الأمر (ACTIVATE POPUP)

ويستخدم لتشغيل عمود قوائم (Menu Bar) مع القوائم المتفرعة منه (POPUP).

٧ - الأمر (ACTIVATE SCREEN)

وهو يؤدي الى العودة الى الشاشة الكاملة بدلا من النافذة المفتوحة (Active Window).

٨ - الأمر (ACTIVATE WINDOW)

ويستخدم فى تشغيل نافذة معينة بدلا من الشاشة الكاملة .والصورة العامة كالآتى :

```
ACTIVATE WINDOW <window name list> / ALL
```

٩ - الأمر (APPEND FROM ARRAY)

ويستخدم فى اضافة سجلات الى ملف قاعدة البيانات من مصفوفة بيانات (Array). والصورة العامة كالآتى :

```
APPEND FROM  ARRAY <array name>
             [ FOR <condition> ]
```

١٠ - الأمر (APPEND MEMO)

ويستخدم في نقل بيانات من ملف نصي (Text File) الى حقل الملاحظات (Memo Field).

١١ - الأمر (ASSIST)

وهو يؤدي الى تشغيل مركز التحكم (Control Center) الخاص ببرنامج (DBase IV) .

١٢ - الأمر (BEGIN TRANSACTION)

ويستخدم فى بدء تعديل بيانات ملف قاعدة البيانات المفتوح بناء على بيانات ملفات الحركة . والصورة العامة كالآتى :

```
BEGIN TRANSACTION [<path name>]
         <transaction commands>
         END TRANSACTION
```

١٣ - الأمر (CALCULATE)

ويستخدم فى عمل حسابات مالية واحصائية للبيانات العددية . والصورة العامة له كالآتى :

```
CALCULATE [ Scope ] <expression list>
                [ FOR <condition> ]
                [ WHILE <condition> ]
                [ TO <memvar list> ]
                [ TO ARRAY <array name> ]
```

حيث (expression list) هي أى عملية حسابية مثل (AVG) أى المتوسط أو (MAX) أى القيمة العظمى أو (MIN) أى القيمة الصغرى أو (STD) أى الانحراف المعيارى أو . . . الخ .

١٤ - الأمر (COMPILE)

وهو يستخدم في تحويل البرنامج المكتوب بشفرة المصدر (Source Code) الى برنامج منفذ (Executable Program) .

١٥ - الأمر (COPY TO ARRAY)

ويستخدم في ملء مصفوفة معينة بالبيانات الموجودة في ملف قاعدة البيانات المفتوح . والصورة العامة له كالآتي :

```
COPY TO ARRAY <array name>
        [ FIELDS <field list>]
        [ FOR <condition>]
        [ WHILE <condition>]
```

١٦ - الأمر (CREATE APPLICATION)

ويستخدم لتشغيل مولد التطبيقات (Application Generator) الخاص ببرنامج (DBase IV) . وهو الذي يساعد على انشاء البرامج التطبيقية بواسطة برنامج (DBase IV) . والصورة العامة له كالآتي :

```
CREATE  / MODIFY APPLICATION <file name> /?
```

١٧ - الأمر (DEACTIVATE MENU)

ويستخدم في مسح عمود القوائم الموجود على الشاشة والاحتفاظ به في الذاكرة حتى يتم استخدامه مع الجمل الاختيارية (ON SELECTION)

١٨ - الأمر (DEACTIVATE POPUP)

ويستخدم في مسح عمود القوائم والقوائم المتفرعة منه مع الاحتفاظ بها في الذاكرة .

١٩ - الأمر (DEACTIVATE WINDOW)

ويستخدم في مسح نوافذ معينة من الشاشة مع الاحتفاظ بها في الذاكرة . والصورة العامة له كالآتي :

```
DEACTIVATE WINDOW <window name list> /ALL
```

٢٠ - الأمر (DEBUG)

ويستخدم لتشغيل برنامج التصحيح (Debugger) الخاص ببرنامج (DBase IV) . والصورة العامة له كالآتي :

```
DEBUG <file name > / <procedure name>
          [ WITH <parameter list>]
```

٢١ - الأمر (DECLARE)

ويستخدم لانشاء مصفوفة أحادية أو ثنائية الأبعاد . والصورة العامة له كالآتي :

```
DECLARE <array name1>
         [ <number of rows> ]
         [ <number of columns> ]
           <array name2>
         [ <number of rows> ]
         [ <number of columns> ] . . . . . ]
```

٢٢ - الأمر (DEFINE BAR)

ويستخدم هذا الأمر لتعريف أحد الاختيارات في عمود القوائم (Menu Bar) والصورة العامة له كالآتي :

```
DEFINE BAR <line number> of <popup name>
          PROMPT <expC>
          [ MESSAGE <expC> ]
          [ SKIP [ FOR <condition> ] ]
```

٢٣ - الأمر (DEFINE BOX)

ويستخدم هذا الأمر فى انشاء مستطيل حول البيانات والصورة العامة له كالآتى :

```
DEFINE BOX
          FROM <point column>
          TO <point column>
          HEIGHT <expN>
          [ AT LINE <point line> ]
          [SINGLE/DOUBLE/<border definition string>]
```

٢٤ - الأمر (DEFINE MENU)

ويستخدم هذا الأمر فى انشاء القوائم (Menus) . والصورة العامة له كالآتى :

```
DEFINE MENU <menu name>
            [MESSAGE <expC>]
```

٢٥ - الأمر (DEFINE PAD)

ويستخدم فى تعريف أحد الاختيارات فى القائمة الفرعية (Popup Menu) والصورة العامة له كالآتى :

```
DEFINE PAD <pad name> OF <menu name>
           PROMPT <expC>
           [AT <row> , <col>]
           [MESSAGE <expC>]
```

٢٦ - الأمر (DEFINE POPUP)

ويستخدم فى تعريف النافذة الفرعية (Popup Window) . والصورة العامة له كالآتى :

```
DEFINE POPUP <popup name>
           FROM <row1> , <col1>
           [ TO <row2> , <col2> ]
[PROMPT FIELD <field name>/PROMPT FILES]
           [ LIKE <skeleton> ]
           [PROMPT STRUCTURE ]
           [ MESSAGE <expC> ]
```

٢٧ - الأمر (DEFINE WINDOW)

ويستخدم فى تعريف نافذة معينة متضمنا الحدود والألوان . والصورة العامة له كالآتى :

```
DEFINE WINDOW <window name>
                 FROM <row1> , <col1>
                 TO <row2> , <col2>
[ DOUBLE / PANEL / NONE / <border definition string> ]
[ COLOR [ <stand> ] [ , <enhanced> ] [ , <frame> ] ]
```

٢٨ - الأمر (LIST/DISPLAY FILES)

ويستخدم لعرض معلومات عن الفهرس الحالى . وهو مثل الأمر (DIR) فى نظام التشغيل (DOS) .

٢٩ - الأمر (LIST/DISPLAY USERS)

ويستخدم لعرض حالة محطات العمل (Work Stations) التى تعمل مع البرنامج من خلال شبكة حاسبات (Network) .

٣٠ - الأمر (LOGOUT)

ويستخدم لفصل احدى المحطات (Work Stations) المتصلة من خلال شبكة (Network).

٣١ - الأمر (MOVE WINDOW)

ويستخدم لتحريك نافذة معينة على الشاشة . والصورة العامة له كالآتى :

```
MOVE WINDWO <window name>
         TO <row> , <column>
         / BY <delta row> , <delta column>
```

٣٢ - الأمر (ON PAD)

ويستخدم لربط قائمة فرعية معينة بعمود اختيارات معين .والصورة العامة له كالآتى :

```
ON PAD <pad name> OF <menu name>
            [ ACTIVATE POPUP <popup name> ]
```

٣٣ - الأمر (ON PAGE)

ويستخدم للتحكم فى ناقل الصفحة (Page Break) . وكذلك فى العناوين العلوية (Headers) والعناوين السفلية (Footers) أثناء طباعة التقارير . والصورة العامة له كالآتى

```
ON PAGE [ AT LINE <expC><command> ]
```

٣٤ - الأمر (ON READERROR)

ويستخدم فى التعامل مع الأخطاء أثناء تشغيل البرنامج . والصورة العامة له كالآتى :

```
ON READERROR [ <command> ]
```

٣٥ - الأمر (ON SELECTION PAD)

ويستخدم فى تنفيذ عمل معين عند اختيار أحد الاختيارات فى احدى القوائم الفرعية . والصورة العامة له كالآتى :

```
ON SELECTION PAD <pad name> OF <menu name>
                 [ <command> ]
```

٣٦ - الأمر (ON SELECTION POPUP)

ويستخدم فى تنفيذ أمر معين عند اختيار أحد القوائم الفرعية (POPUP). والصورة العامة له كالآتى :

```
ON SELECTION POPUP <popup name>
           / ALL [ <command> ]
```

٣٧ - الأمر (PLAY MACRO)

ويستخدم في تنفيذ مجموعة من الأوامر التي سبق تخزينها فى مجموعة من المفاتيح (Macro) . والصورة العامة له كالآتى :

```
PLAY MACRO <macro name>
```

٣٨ - الأمر (PRINTJOB)

ويستخدم لتنفيذ بعض الأوامر أثناء تنفيذ عملية الطباعة . والصورة العامة له كالآتى :

```
PRINTJOB
        <commands>
ENDPRINTJOB
```

٣٩ - الأمر (PROTECT)

ويستخدم فى تأمين بعض البيانات وحمايتها من الاستخدام الا بواسطة المختصين .

٤٠ - الأمر (RELEASE MODULE)

ويستخدم في مسح أى برنامج فرعي أو نافذة أو قائمة أو برنامج مكتوب بلغة التجميع من الذاكرة المؤقتة . والصورة العامة له كالآتى :

```
RELEASE MODULE [ <module name list> ]
            / MENUS [ <menu name list> ]
            / POPUPS [<POPU name list> ]
            / WINDOWS [ <windows name list> ]
```

٤١ - الأمر (RESTORE MACRO)

ويستخدم لتحميل برنامج حروف مبرمجة (Macro) فى الذاكرة المؤقتة . والصورة العامة له كالآتى :

```
RESTORE MACRO FROM <macro file>
```

٤٢ - الأمر (RESTORE WINDOW)

ويستخدم لتحميل نافذة من القرص الى الذاكرة المؤقتة .والصورة العامة له كالآتى :

```
RESTORE WINDOW <window name list>
                   / ALL FROM <file name>
```

٤٣ - الأمر (ROLLBACK)

ويستخدم لاعادة ملف قاعدة البيانات الى حالته الأولى قبل ادخال حركة معينة (Transaction). والصورة العامة له كالآتى :

```
ROLLBACK [<database filename>]
```

٤٤ - الأمر (SAVE MACROS)

ويستخدم لتخزين برنامج الفاتيح المبرمجة (Macro) فى ملف معين . والصورة العامة له كالآتى :

```
SAVE MACROS TO <macro file>
```

٤٥ - الأمر (SAVE WINDOW)

ويستخدم لتخزين نافذة معينة فى ملف .

٤٦ - الأمر (SCAN)

ويستخدم لتنفيذ بعض الأوامر على مجموعة من السجلات . والصورة العامة له كالآتى :

```
SCAN
        [ <scope>]
        [FOR <condition>]
        [ WHILE <condition>]
        [ <commands>]
        [ LOOP]
        [ EXIT]
ENDSCAN
```

٤٧ - الأمر (SHOW MENU)

ويستخدم لعرض قائمة على الشاشة دون تشغيلها . والصورة العامة لـــــه كالآتى :

```
SHOW MENU <menu name>
          [PAD <pad name>]
```

٤٨ - الأمر (SHOW POPUP)

ويستخدم لعرض قائمة فرعية على الشاشة دون تشغيلها . والصورة العامة له كالآتى :

```
SHOW POPUP <popup menu>
```

٤٩ - الأمر (UNLOCK)

ويستخدم لفتح الملف أو السجل حتى يصبح قابلا للتشغيل بواسطة المستخدم . والصورة العامة له كالآتى :

```
UNLOCK   [ALL / IN <alias>]
```

ثانيا : أوامر التجهيز (SET)

١ - الأمر (SET)

ويستخدم فى عرض قوائم الضبط .

٢ - الأمر (SET AUTOSAVE)

ويستخدم فى تخزين السجلات من المخزن المؤقت (Buffer) الى القرص . والصورة العامة له كالآتى :

```
SET AUTOSAVE ON/OFF
```

٣ - الأمر (SET BELL TO)

ويستخدم في التحكم فى الصوت الذى يحدث عند تشغيل الجرس . والصورة العامة له كالآتى :

```
SET BELL TO [<frequency>,<duration>]
```

٤ - الأمر (SET BORDER TO)

ويستخدم فى التحكم فى حدود النوافذ والقوائم الفرعية . والصورة العامة له كالآتى :

```
SET BORDER TO
    [SINGLE/DOUBLE/PANEL/NONE
    / <border definition string>]
```

٥ - الأمر (SET CLOCK)

ويستخدم في عرض ساعة النظام فى الركن العلوى الأيمن من الشاشة . والصورة العامة له كالآتى :

```
SET CLOCK ON/OFF
```

٦ - الأمر (SET CLOCK TO)

ويستخدم فى التحكم فى مكان عرض ساعة النظام . والصورة العامة لـــه كالآتى :

```
SET CLOCK TO [<row> , <column>]
```

٧ - الأمر (SET COLOR OF)

ويستخدم فى تحديد ألوان أشياء محددة على الشاشة .والصورة العامة له كالآتى :

```
SET CLOCR OF NORMAL/MESSAGES
          /TITLES/HIGHLIGHT/BOX
          /INFORMATION/FIELDS
          TO [<attribute>]
```

٨ - الأمر (SET CURRENCY)

ويستخدم لعرض علامة العملة على اليمين أو اليسار حسب الحاجة .والصورة العامة له كالآتى :

```
SET CURRENCY LEFT/RIGHT
```

٩ - الأمر (SET DATE)

ويستخدم فى تحديد طرق كتابة التاريخ طبقا للدول المختلفة .والصورة العامة له كالآتى :

```
SET DATE [TO] AMERICAN/ANSI/BRTISH
    /FRENCH/GERMAN/ITALIAN/JAPAN/USA
    /MDY/DMY/YMD
```

١٠ - الأمر (SET DEVELOPMENT ON/OFF)

ويستخدم عند تطوير البرنامج وعمل بعض التعديلات ، فعندما تكون شفرة المصدر (Source Code) قد تـــم تعديلـــها بتاريخ يـــلى تاريـــخ شفرة الهدف (Object Code) يقوم برنامج (DBase IV) باعادة ترجمة البرنامج (Compiling) .

١١ - الأمر (SET DISPLAY TO)

ويستخدم لضبط البرنامج على حالة الشاشة (Display Mode) اذا كانت (Mono) او (EGA) أو (VGA) . والصورة العامة له كالآتى :

```
SET DISPLAY TO MONO/COLOR/EGA 25
        /EGA 43/ MONO 43
```

١٢ - الأمر (SET ENCRYPTION)

ويستخدم فى تشفير أو فك شفرة البرنامج الذى سبق حمايته بواسطة الأمر (PROTECT) . والصورة العامة له كالآتى :

```
SET ENCRYPTION ON/OFF
```

١٣ - الأمر (SET HOURS TO)

ويستخدم في عرض الساعة وتحديد عدد الساعات بها اذا كان (١٢) أو (٢٤) ساعة . والصورة العامة له كالآتى :

```
SET HOURS TO [12/24]
```

١٤ - الأمر (SET LOCK)

ويستخدم لتشغيل أو عدم تشغيل قفل السجل ، وهذا يعني التحكم في قدرة المستخدم على التعامل مع بيانات هذا السجل أو عدم قدرته على ذلك .والصورة العامة له كالآتى :

```
SET LOCK TO [<expC>]
```

١٥ - الأمر (SET MARK TO)

ويستخدم لتحديد الحرف الذى يفصل أرقام التاريخ ، والحرف المبدئى هو الحرف (/) .

١٦ - الأمر (SET PAUSE)

ويستخدم فى ايقاف الشاشة أو عدم ايقافها عند عرض بيانات تزيد عن طول الشاشة . والصورة العامة له كالآتى :

```
SET PAUSE ON/OFF
```

١٧ - الأمر (SET POINT TO)

ويستخدم لتحديد الحرف الذي يمثل العلامة العشرية . والحرف المبدئى هو النقطة (.) . والصورة العامة له كالآتى :

```
SET POINT TO [<expC>]
```

١٨ - الأمر (SET PRECISION)

ويستخدم لتحديد عدد الكسور العشرية التى تظهر فى الأعداد . والعدد المبدئى هو (١٦) والمدى من (١٠) الى (٢٠) . والصورة العامة له كالآتى :

```
SET PRECISION TO [<expC>]
```

١٩ - الأمر (SET PRINTER)

ويستخدم فى توجيه أو عدم توجيه المخرجات الى الطابعة . والصورة العامة له كالآتى :

```
SET PRINTER ON/OFF
```

٢٠ - الأمر (SET PRINTER TO)

ويستخدم فى توجيه المخرجات الى طابعة محددة . والطابعة المبدئية هى (PRN) . والصورة العامة له كالآتى :

```
SET PRINTER TO <computer name>
               <printer name> = <destination>
```

٢١ - الأمر (SET PRINTER TO FILE)

ويستخدم لتوجيه المخرجات الى ملف بدلا من الطابعة .والصورة العامة له كالآتى :

SET PRINTER TO FILE <file name>

٢٢ - الأمر (SET WINDOW OF MEMO)

ويستخدم فى تحديد النافذة التـــى يتـــم عـــن طريقها تعديل حقول الملاحظات (Memo Fields) . والصورة العامة له كالآتى :

SET WINDOW OF MEMO TO <window name>

ثالثا : الدوال

١ - الدالة (ACCESS)

وهى دالة يتم عن طريقها تحديد مستويات التعامل مع البيانات للمستخدمين حسب درجة السرية الخاصة بكل منهم .

٢ - الدالة (ACOS)

وهى دالة تحسب مقدار الزاوية من جيب تمامها .

٣ - الدالة (ALIAS)

وهى دالة تحدد الاسم المرادف لمنطقة عمل محددة .

٤ - الدالة (ASIN)

وهى دالة تحسب مقدار الزاوية من جيبها .

٥ - الدالة (ATAN)

وهى دالة تحسب مقدار الزاوية من ظلها .

٦ - الدالة (ATN2)

وهى دالة تحسب مقدار الزاوية من جيبها وجيب تمامها . والصـــورة العامة له كالآتى :

```
ATN2 (<expN1> , <expN2>)
```

٧ - الدالة (BAR)

وهى دالة تعطى رقم آخر عمود قوائم (Menu Bar) ثم اختياره .

٨ - الدالة (CEILING)

وهى دالة تحدد أصغر قيمة تزيد عن أو تساوى القيمة المحددة فى الدالة . والصورة العامة له كالآتى :

```
CEILING (<expN>)
```

٩ - الدالة (COMPLETED)

وهى دالة تحدد ما اذا كانت عملية ادخال الحركة (Transaction) قد تمت أم لا .

١٠ - الدالة (COS)

وهى دالة تحسب جيب تمام الزاوية .

١١ - الدالة (DIFFERENCE)

وهى تعطى الفرق بين سلسلتين حرفيتين . والصورة العامة لها كالآتى :

```
DIFFERENCE (<expC> , <expC>)
```

١٢ - الدالة (DTOR)

وهى دالة تقوم بتحويل الزاوية من الدرجات (Degrees) الى الوحدات القطرية (Radians).

١٣ - الدالة (FILE)

وهى دالة تتحقق من وجود ملف معين .

١٤ - الدالة (FKLABEL)

وهى تحدد اسم مفتاح وظيفة معين (Function Key) من رقمه . والصورة العامة له كالآتى :

```
FKLABEL (<expN>)
```

١٥ - الدالة (FKMAX)

وهى دالة تحدد أكبر عدد من مفاتيح الوظائف يمكن برمجته .

١٦ - الدالة (FLOCK)

وهى تستخدم فى اغلاق ملف قاعدة البيانات حتى لا يستطيع أى شخص التعامل معه .

١٧ - الدالة (FLOOR)

وهى دالة تحدد أكبر عدد أصغر من أو يساوى القيمة المحددة فى الدالة .

١٨ - الدالة (FV)

وهى دالة تعطى القيمة المستقبلية (Future Value) لاستثمار معين لفترة محددة وبمعدل ثابت . والصورة العامة له كالآتى :

```
FV  (<payment> , <rate> , <period>)
```

١٩ - الدالة (LASTKEY)

وهى دالة تعطى كود الآسكى (ASCII Code) لآخر مفتاح تم الضغط عليه للخروج من شاشة معينة .

٢٠ - الدالة (LIKE)

وهى دالة تقوم بمقارنة سلسلتين حرفيتين . والصورة العامة له كالآتى :

```
LIKE   (<pattern> , <expC>)
```

٢١ - الدالة (LINNO)

وهى تحدد رقم السطر الذى سوف يتم تنفيذه من البرنامج .

٢٢ - الدالة (LOCK)

وتستخدم فى اغلاق سجلات معينة من ملف قاعدة البيانات خلال شبكة الحاسبات (Network) حتى لا تستخدم بواسطة أى مستخدم .والصورة العامة له كالآتى :

```
LOCK   ([<exp list>] [,<alias>])
```

٢٣ - الدالة (MENU)

وهى دالة تحدد اسم القائمة الفرعية المفتوحة .

٢٤ - الدالة (MESSAGE)

وهى دالة تحدد رسالة الخطأ الخاصة بآخر خطأ ظهر فى البرنامج .

٢٥ - الدالة (MLINE)

وهى دالة تحدد سطرا معينا فى حقل الملاحظات .

٢٦ - الدالة (NETWORK)

وهى دالة تحدد ما اذا كان البرنامج يعمل على الشبكة (Network) أم لا .

٢٧ - الدالة (PAD)

وهى تحدد اسم القائمة الجارى العمل عليها .

٢٨ - الدالة (PAYMENT)

وهى دالة تحدد الأقساط المدفوعة لسداد قرض معين فى فترة معينة وبفائدة معينة . والصورة العامة له كالآتى :

```
PAYMENT (<principal> , <rate> , <period>)
```

٢٩ - الدالة (PI)

وهى دالة تعطى النسبة التقريبية (ط) .

٣٠ - الدالة (POPUP)

وهى دالة تعطى اسم القائمة الفرعية المفتوحة .

٣١ - الدالة (PRINTSTATUS)

وهى دالة تعطى حالة الطابعة الموصلة .

٣٢ - الدالة (PROGRAM)

وهى دالة تعطى اسم البرنامج الجارى تنفيذه عندما يظهر خطأ معين . وذلك لتحديد البرنامج الفرعى المحتوى على الخطأ .

٣٣ - الدالة (PV)

وهى تعطى القيمة الحالية لقرض يتم تسديده على أقساط فى فترات زمنية محددة . والصورة العامة له كالآتى :

```
PV (<payment> , <rate> , <period>)
```

٢٤ - الدالة (RAND)

وهى دالة تقوم بتوليد أعداد عشوائية .

٢٥ - الدالة(RTOD)

وهى تستخدم لتحويل الزوايا من التقدير الدائرى(Radian) الى درجات .

ملحق (٢)

أهم الأوامر والدوال المستخدمة

فى برنامج (FoxBase+)

هذا الملحق يضم الاوامر والدوال الاضافية فقط . حيث ان باقى الاوامر و الدوال مطابقة تماما لما سبق عرضه فى برامج (+DBaseIII) ، (DBaseIV) .

أولا : الأوامـــر

١ - الأمر (CLEAR PROGRAM)

ويستخدم فى مسح برنامج مكتوب بلغة التجميع (Assembly) .

٢ - الأمر (CLEAR TYPEAHEAD)

ويستخدم لمسح مخزن الذاكرة المؤقت (Buffer) .

٣ - الأمر (DELETE FILE)

ويستخدم فى مسح ملف معين من القرص .

٤ - الأمر (DIMENSION)

ويستخدم فى انشاء مصفوفة متغيرات أحادية أو ثنائية .

٥ - الأمر (DIR)

ويستخدم فى عرض فهرس الملفات .

٦ - الأمر (FLUSH)

ويستخدم فى نقل محتويات الذاكرة الى القرص .

٧ - الأمر (GATHER FROM)

ويستخدم فى تخزين بيانات المصفوفة فى متغيرات ذاكرة . والصورة العامة له

كالآتى :

```
GATHER FROM <array> [FIELDS <fields>]
```

٨ - الأمر (MENU)

ويستخدم لتشغيل عمود اختيارات (Menu Bar) .

٩ - الأمر (ON KEY = <expN>)

ويستخدم فى تنفيذ مجموعة من الأوامر عند ضغط المستخدم على مفتاح محدد .

١٠ - الأمر (RELEASE MODULE)

ويستخدم فى مسح برنامج سبق تحميله فى الذاكرة .

١١ - الأمر (RESTORE SCREEN)

ويستخدم لتحميل شاشة من الذاكرة أو من متغيرات ذاكرة معينة .

١٢ - الأمر (SAVE SCREEN)

ويستخدم لتخزين الشاشة فى متغير ذاكرة محدد .

١٣ - الأمر (SCATTER)

ويستخدم فى نقل محتويات متغيرات ذاكرة الى مصفوفة متغيرات . والصورة العامة له كالآتى :

```
SCATTER [FIELDS <fields list> ] TO <array>
```

١٤ - الأمر (UNLOCK)

ويستخدم فى مسح كل أقفال الملفات والسجلات حتى يمكن التعامل معها بواسطة أي مستخدم .

ثانيا : الدوال

١ - الدالة (ALIAS)

وهى تعطى الاسم المرادف لمنطقة العمل المفتوحة .

٢ - الدالة (FCOUNT)

وهى تعطى عدد الحقول الموجودة فى ملف قاعدة البيانات المفتوح .

٣ - الدالة (FILE)

وهى تختبر وجود ملف معين على القرص .

٤ - الدالة (FKLABEL)

وهى تعطى اسم مفتاح الوظيفة (Function Key) القابل لرقم معين .

٥ - الدالة (FKMAX)

وهى تعطى أكبر عدد يمكن برمجته من مفاتيح الوظائف .

٦ - الدالة (FLOCK)

وهى تستخدم فى اغلاق ملف قاعدة بيانات محدد حتى لايتم استخدامه بواسطة باقى المستخدمين .

٧ - الدالة (LOCK)

وهى تستخدم فى اغلاق السجل المفتوح حتى لايتم استخدامه بواسطة المستخدمين الآخرين .

٨ - الدالة ((SYS(0)

وهى تعطى اسم الجهاز ورقمه فى شبكة الحاسبات .

٩ - الدالـة ((SYS(1)

وهى تعطى التاريخ بالشكل المستخدم فى برنـامج (FoxBase) وبـرنـامـج (DBase II) .

١٠ - الدالـة ((SYS(2)

وهى تعطى عدد الثوانى المنقضية منذ منتصف الليل وحتى الوقت الحـالى ، أى أنها تعطى الوقت الحالى .

١١ - الدالـة ((SYS(3)

وهى تعطى اسم ملف مؤقت يتم استخدامه بواسطة أحد المستخدمين المتـصلـين بالشبكة .

١٢ - الدالـة ((SYS(5)

وهى تعطى اسم وحدة الأقراص المبدئية (Default) .

١٣ - الدالـة ((SYS(6)

وهى تعطى الطابعة الحالية (Current Printer) .

١٤ - الدالـة ((| W | ، SYS(7)

وهى تعطى اسم ملف التشكيل المفتوح (Format File) فى منطقة العمـل الممثلة بالرقم (W) .

١٥ - الدالة (SYS(9))

وهى تعطى رقم نسخة برنامج (FoxBase +) .

١٦ - الدالة (SYS(10,d))

وهى تحول الرقم الممثل للتاريخ (d) الى تاريخ حرفى (Character Date) مطابق فى الشكل للتاريخ المستخدم بواسطة برنامج (FoxBase) وبرنامج (DBase II) .

١٧ - الدالة (SYS(11,s))

وهى عكس الدالة السابقة ، حيث تعطى الرقم الممثل للتاريخ الحرفى (s) .

١٨ - الدالة (SYS(12))

وهى تعطى الذاكرة المؤقتة المتاحة .

١٩ - الدالة (SYS(13))

وهى تعطى حالة الطابعة اذا كانت جاهزة أو غير جاهزة .

٢٠ - الدالة (SYS(14,n ,w))

وهى تعطى الملف الفهرسى رقم (n) فى منطقة العمل (W) مع ملاحظة أن العدد (n) لايزيد عن (٧) والعدد (W) لايزيد عن (١٠) .

٢١ - الدالة (SYS(15,t,s))

وهى دالة تعطى السلسلة الحرفية (s) المقابلة للسلسلة الحرفية (t) حسب جدول التحويل الخاص بالمستخدم (Translation Table) .

٢٢ - الدالة (SYS(16,n))

وهى دالة تعطى اسم البرنامج الجارى تنفيذه ، وهى تفيد عند متابعة خطأ

معين ومحاولة التوصل الى البرنامج المتسبب فى هذاالخطأ . والعدد (n) يمثل المستوى الذى يقع عليه البرنامج ، فالعدد (١) يمثل البرنامج الرئيسى (Master Program) ، واذا تم حذف العدد (n) فان الدالة تعطى البرنامج الحالي الجارى تنفيذه .واذا أريد متابعة جميع البرامج الفرعية يتم استخدام الأوامر التالية :

```
STORE 1 TO i
DO WHILE LEN(SYS(16,i)<>0)
     ? SYS(16,i)
     STORE i+1 TO i
ENDDO
```

٢٣ - الدالة (SYS(17))

وهى تعطى نوع المعالج الدقيق المستخدم فى الجهاز اذا كان (8086/8088) أو (80286) او (80386) .

٢٤ - الدالة (SYS(18))

وهى تعطى اسم حقل قاعدة البيانات الذى يتم ادخاله عن طريق الأمر (@...GET) ، وذلك عند ضغط المستخدم على المفتاح الذى تم تحديده بواسطة الأمر (ON KEY=t) حيث (t) يمثل اسم المفتاح .

٢٥ - الدالة (SYS(100))

وهى تعطى حالة الشاشة (Console) اذا كانت (ON) أو (OFF) .

٢٦ - الدالة (SYS(101))

وهى تعطى حالة الجهاز المستخدم اذا كان الشاشة أو الطابعة .

٢٧ - الدالة (SYS(102))

وهى تعطى حالة الطابعة (Printer) اذا كانت (ON) أو (OFF) .

٢٨ - الدالة ((SYS(103)

وهى تعطى حالة الأمر (SET TALK) اذا كانت (ON) أو (OFF) .

٢٩ - الدالة (UPDATED)

وهى تعطى حالة البيانات اذا كان قد تم تحديثها أم لا .

٣٠ - الدالة (VERSION)

وهى تعطى رقم نسخة برنامج (+ FoxBase) المستخدم .

ملحق (٣)

أهم الأوامر و الدوال المستخدمة

فى برنامج (FoxPro)

هذا الملحق يضم الأوامر والدوال الأضافية فقط ، حيث أن باقي الأوامر والدوال مطابقة تماما لما سبق عرضه في برامج (+ FoxBase) ، (DBase IV) ، (DBase III +) .

أولا : الأوامـــر

١ - الأمـــر (COPY MEMO)

ويستخدم لنسخ محتويات حقل الملاحظات في ملف .

٢ - الأمر (FILER)

ويستخدم في تشغيل برنامج صيانة الملفات الخاص ببرنامج (FoxPro) .

٣ - الأمر (FOR)

ويستخدم في انشاء حلقة تكرارية مع تشغيل عداد لحساب عدد مرات تنفيذ الحلقة . والصورة العامة له كالآتي :

```
FOR <memory>=<expN1>TO<expN2>[STEP<expN3>]<statements>
              [EXIT]
              [LOOP]
        ENDFOR
```

٤ - الأمـــر (FUNCTION)

وهو أمر يحدد بداية برنامج فرعي (Subroutine). والصـــورة العامة له كالآتي :

```
FUNCTION <name>
```

٥ - الأمـــر (MODIFY MEMO)

ويستخدم في تعديل محتويات حقل الملاحظات .

٦ - الأمـــر (SET COLOR SET)

ويستخدم فى تحميل مجموعة من الألوان سبق تحديدها .

٧ - الأمـــر (SET CURSOR)

ويستخدم فى التحكم فى مؤشر الشاشة .

٨ - الأمـــر (SET LOGERRORS)

ويستخدم فى ارسال أخطاء البرنامج أثناء ترجمته الى ملف على القرص .

٩ - الأمر (SET MACKEY TO)

ويستخدم فى تحديد مفتاح أو عدة مفاتيح لتشغيل برنامج الماكرو .

١٠ - الأمـــر (SET MOUSE)

ويستخدم فى تشغيل الفأرة (Mouse) .

١١ - الأمـــر (SET MOUSE TO)

ويستخدم فى التحكم فى الفأرة (Mouse) .

١٢ - الأمـــر (SET RELATION OFF INTO)

ويستخدم فى الغاء علاقة بين ملفين مفتوحين .

١٣ - الأمـــر (SET SHADOWS)

ويستخدم فى التحكم فى ظهور ظل النافذة أو عم ظهوره .

ثانيا : الدوال

١ - الدالـة (ALLTRIM)

وتستخدم فى حذف المسافات الخالية من أول السلسلة الحرفية وآخرها .

٢ - الدالـة (ATC)

وتستخدم فى البحث عن سلسلة حرفية داخل سلسلة حرفية أخرى بصرف النظر عن حالة الحروف اذا كانت صغيرة أو كبيرة ، وهى تعطى عددا يمثل مكان هذه السلسلة . والصورة العامة له كالآتى :

```
ATC   (<expC1> , <expC2>[ , <expN>])
```

٣ - الدالـة (ATCLINE)

وهى تبحث عن سلسلة حرفية معينة داخل سلسلة حرفية أخرى بصرف النظر عن حالة الحروف اذا كانت صغيرة أو كبيرة ، وهى تعطى رقم السطر المحتوى على هذه السلسلة . والصورة العامة له كالآتى :

```
ATCLINE  (<expC1> , <expC2>)
```

٤ - الدالـة (ATN2)

وهى تعطى قيمة الزاوية بالتقدير الدائرى من جيب الزاوية وجيب تمامها . والصورة العامة له كالآتى :

```
ATN2  (<expN1> , <expN2>)
```

٥ - الدالـة (CAPSLOCK)

وهى تعطى حالة مفتاح الحروف الكبيرة (Caps) .

٦ - الدالـة (CHRSAW)

وهى تحدد ما اذا كان أحد الحروف موجودا فى مخزن الذاكرة الخاص بلوحة المفاتيح .

٧ - الدالـة (CHRTRAN)

وهى دالة تقوم بترجمة حرف أو حروف معينة باستخدام جداول ترجمة محددة (Translation Tables) . والصورة العامة له كالآتى :

```
CHRTRAN  (<expC1> , <expC2> , <expC3>)
```

٨ - الدالـة (CURDIR)

وهى تعطى الفهرس الحالى (Current Directory) .

٩ - الدالـة (DATE)

وهى تعطى تاريخ اليوم الحالى .

١٠ - الدالـة (DMY)

وهى دالة تقوم بتحويل التاريخ الى الشكل (Day Month Year) .

١١ - الدالـة (EMPTY)

وهى تحدد اذا كانت السلسلة خالية (Blank) أم لا .

١٢ - الدالـة (FCOUNT)

وهى تعطى عدد الحقول فى ملف قاعدة البيانات المفتوح .

١٣ - الدالـة (FGETS)

وهى تعطى حروفا من ملف حتى يتم الضغط على مفتاح الادخال .

١٤ - الدالـة (FILTER)

وهى تعطى الشروط المستخدمة فى المرشح (Filter) المستخدم .

١٥ - الدالـة (FOPEN)

وتستخدم فى فتح ملف .

١٦ - الدالـة (FPUTS)

وهى تقوم بكتابة سلسلة حرفية فى الملف المفتوح .

١٧ - الدالـة (FREAD)

وهى تقوم بقراءة عدد من الحروف من الملف المفتوح .

١٨ - الدالـة (FSIZE)

وهى تعطى حجم حقل معين بالحروف (Bytes) .

١٩ - الدالـة (FULLPATH)

وهى تعطى المسار الخاص بملف معين .

٢٠ - الدالـة (FWRITE)

وهى تكتب سلسلة حرفية معينة فى الملف المفتوح .

٢١ - الدالـة (HEADER)

وهـــى تعطى عدد الحروف الموجودة فى العنوان (Header) الخاص بأحد الملفات .

٢٢ - الدالـة (INLIST)

وهى تحدد ما اذا كانت سلسلة حرفيه معينة موجودة ضمن مجموعة من السلاسل الحرفية .

٢٣ - الدالـة (ISDIGIT)

وهى تحدد ما اذا كان أول حرف من سلسلة حرفية رقما أم حرفا .

٢٤ - الدالـة (MCOL)

وهى تعطى مكان العمود الخاص بمؤشر نافذة معينة .

٢٥ - الدالـة (MDY)

وهى تحول التاريخ الى صورة (شهر يوم سنة)أى(Month, Day, Year).

٢٥ - الدالـة (MEMLINES)

وهى تعطى عدد السطور فى حقل الملاحظات .

٢ - الدالـة (MEMORY)

وهى تعطى الذاكرة المؤقتة المتاحة .

٢٧ - الدالـة (MESSAGE)

وهى تعطى رسالة الخطأ المستخدمة .

٢٨ - الدالـة (MROW)

وهى تعطى رقم العمود الخاص بمؤشر نافذة معينة .

٢٩ - الدالـة (OCCURS)

وهي تعطي عدد مرات وجود سلسلة حرفية معينة داخل سلسلة حـرفـية أخـرى . والصورة العامة له كالآتى :

```
OCCURS  (&lt;expC1&gt; , &lt;expC2&gt;)
```

٣٠ - الدالـة (ORDER)

وهى تعطى اسم الفهرس الرئيسى فى منطقة عمل محددة .

٣١ - الدالـة (PARAMETERS)

وهى تعطى عـدد المعاملات التى تـم ادخـالهـا الـى آخـر بـرنـامـج خـطـوات .(Procedure)

٣٢ - الدالـة (PROPER)

وهي تعطى الأسماء بشكلها المتعارف عليه . وهـو أن يكـون أول حـرف كـبـيـرا (Capital) وباقى الحروف صغيرة (Small) .

٣٣ - الدالـة (RATLINE)

وهي تبحث عن وجود سلسلة حرفية معينة داخل سلسلة حرفية أخرى وتحدد رقم السطر الموجودة به . والصورة العامة له كالآتى :

```
RATLINE  (&lt;expC1&gt; , &lt;expC3&gt;)
```

٣٤ - الدالـة (RELATION)

وهى تعطى العلاقة التى سبق تكوينها فى منطقة العمل المفتوحة .

٣٥ - الدالـة (RTOD)

وهى تحول الزاوية من التقدير الدائرى الى درجات .

٣٦ - الدالـة (SCOLS)

وهى تعطى عدد الأعمدة المتاحة على الشاشة .

٣٧ - الدالـة (SIGN)

وهى تحدد الاشارة الخاصة بالعدد سواء كانت سالبة أو موجبة .

٣٨ - الدالة (SROWS)

وهى تعطى عدد السطور المتاحة على الشاشة .

٣٩ - الدالـة (SYS(23))

وهى تعطى حجم الذاكرة الممتدة (Extended) المستخدمة فى برنامج (FoxPro) .

٤٠ - الدالـة (SYS(24))

وهى تعطى حجم الذاكرة الممتدة (Extended) المتاح والذى سبق تحديده من خلال ملف المواصفات (CONFIG.SYS) .

٤١ - الدالة (SYS(1001))

وهى تعطى حجم الذاكرة المتاحة لبرنامج (FoxPro) .

٤٢ - الدالـة (SYS(2000,<expC>))

وهى تعطى اسم الملف المطابق للسلسلة الحرفية (<expC>) .

٤٣ - الدالـة (SYS(2001,<expC>))

وهى تعطى الحالة الحالية (Current Status) للأمر (SET) المثل بالسلسلة الحرفية (<expC>) .

٤٤ - الدالـة (SYS(2002))

وهى تتحكم فى تشغيل أو عدم تشغيل مؤشر الشاشة (Cursor) .

٤٥ - الدالـة (SYS(2003))

وهى تعطى اسم الفهرس الحالى المستخدم .

٤٦ - الدالـة (SYS(2005))

وهى تعطى اسم ملف الموارد المستخدم (Resource File) .

٤٧ - الدالـة (SYS(2006)]

وهى تعطى نوع الشاشة المستخدمة والكارت المستخدم معها .

٤٨ - الدالـة (SYS(2007,<expC>))

وهى تعطى نتيجة اختبار الجمع (Check Sum) للسلسلة الحرفية < expC> .

٤٩ - الدالـة (SYS(2008))

وهى تحدد شكل مؤشر التصحيح المستخدم .

٥٠ - الدالة ((SYS(2009))

وهي تحول بين وضع الاضافة (Insert) ووضع الكتابة الفوقية (Overwrite).

٥١ - الدالة (WCOLS)

وهى تعطى عدد الأعمدة فى نافذة معينة .

٥٢ - الدالة (WEXIST)

وهى تحدد ما اذا كانت نافذة معينة قد تم تعريفها أم لا .

٥٣ - الدالة (WLCOL)

وهى تعطى رقم العمود الخاص بنافذة معينة .

٥٤ - الدالة (WLROW)

وهى تعطى رقم الصف الخاص بنافذة معينة .

٥٥ - الدالة (WOUTPUT)

وهى تحدد ما اذا كانت المخرجات قد تم توجيهها الى نافذة معينة أم لا .

٥٦ - الدالة (WROWS)

وهى تعطى عدد السطور فى نافذة معينة .

موسوعة "دلتا كمبيوتر" لتكنولوجيا وعلوم الحاسب

١ - الحاسبات الالكترونية حاضرها ومستقبلها

٢ - الموسوعة الشاملة لمصطلحات الحاسب الالكترونى

٣ - المرجع الشامل لنظام التشغيل (DOS)

MS DOS
MS WINDOWS NORTON UTILITIES
PC TOOLS VIRUS-SCAN

٤ - عالم الجداول الالكترونية

LOTUS 123
EXCEL
QUATRO PRO

٥ - نظم ادارة قواعد البيانات (الجزء الاول)

FOXBASE+ DBASE III+
FOXPRO DBASE IV

٦ - نظم ادارة قواعد البيانات (الجزء الثانى)

٧ - تطبيقات نظم ادارة قواعد البيانات

٨ - فيروسات الحاسب وأمن البيانات

٩ - النظم المحاسبية والحاسب الالكترونى

حسابات المخازن - الحسابات العامة
حسابات العملاء - حسابات الموردين
حسابات المرتبات

١٠ - الحاسب الالكترونى وادارة المشروعات

١١ - النظم الخبيرة والذكاء الاصطناعى

تعتبر المكتبة العربية ومحتوياتها فى مجال التكنولوجيا من أكبر الدعائم الاساسية للمعرفة والتى تشكل بدورها أحد العوامل الرئيسية لجوانب التنمية المختلفة فى المنطقة العربية . ولما كانت تكنولوجيا الحاسبات من أهم اتجاهات المعرفة التكنولوجية فى الآونة الأخيرة فان قيمة المؤلفات تزداد فى هذا الجانب من واقع ازدياد حاجة المستخدم العربى اليها . ومما لاشك فيه أن المكتبة العربية فى مجال تكنولوجيا وعلوم الحاسب تعتبر فقيرة فى هذا النوع من المؤلفات إلى درجة بعيدة نظرا لعدة جوانب نذكر منها مايلى :

- العمق الفنى اللازم والمواكب للتطور التكنولوجى السريع .
- افتقار المكتبة العربية الى القدر المطلوب من البعد العلمى اللازم للبعد الفنى .
- الترابط الكامل بين جوانب المعرفة فى المراجع المختلفة وعلاقات ذلك بدرجة استفادة القارئ وانعكاسه على درجة المعرفة ومستوى الخبرة .
- درجة ارتباطها بالتطبيق ومستوى استفادة القارئ منها .
- التغطية الكاملة لكل مستويات القراء مع اختلاف ثقافاتهم وخبراتهم .
- حاجة القارئ العربى فى هذه المرحلة لتجاوز مستوى العديد من المراجع المتاحة والتى تعتمد على الترجمة الحرفية لدليل التشغيل للنظم التكنولوجية المختلفة الخاصة بالحاسب.

ومن هذا المنطلق فقد قامت مؤسسة دلتا باعداد موسوعة " دلتا كمبيوتر " لتكنولوجيا وعلوم الحاسب - والتى تتكون من العديد من المراجع - على أيدى نخبة مختارة من أساتذة الجامعات وكبار الخبراء المتخصصين فى هذا المجال .

ومع التطور السريع فى عالم تكنولوجيا الحاسبات وتعدد جوانب المعرفة المطلوبة للقارئ العربى فان موسوعة دلتا قد تم اعدادها على اساس التغطية الشاملة لاتجاهات التكنولوجيا الحديثة تبعا للأولويات المطروحة مع التغطية المستمرة للمستجدات فى هذا المجال من خلال الاصدارات المختلفة لكتب الموسوعة على ضوء التطور السريع فى مجال تكنولوجيا الحاسبات .

دلتا كمبيوتر
Delta Computer

٥ عبد الله بن الزبير مصر الجديدة – نزهة – القاهرة – ت ٢٤٦٧٣٣٨ / ٢٤٤٠٣٧٥ / ٢٤٤٤٩٩٧ فاكسميلى ٢٤٣٥٦٦٥
5 ABDULLAH EBN EL-ZUBAIR NOZHA HELIOPOLIS, CAIRO Fax 2435665 Tel 2467338/2440375/2444997